DZb 83/1633

9/85

TIPS + Tricks von CPU

Neu:
ab sofort
92 Seiten tolle
Tips + Tricks

6,– DM
6,– sFr
50,– öS
18,– FMK
3. Jahrgang

Spezial-Doppelausgabe Sept./Okt.

Neue Serie:
Jetzt wird der Commodore zum universellen Steuerungs-computer

Komfortables Textprogramm gesucht?
Hier eine Version für alle MSX-Besitzer

Commodore 64
Umfangreiche Befehls-erweiterung u.v.m.

Wohin mit Maschinensprache?
Für alle CPC-464-Freunde

Service-Katalog
alle Tips + Tricks auf Datenträger erhältlich

Neu:
Autoren-wettbewerb

VC-64 * CPC-464 * VC-20 * Atari * Apple * TI-99 * ZX-Spectrum * MSX *

Ihr Erfolgsprogramm – aus dem Tronic-Verlag

Compute mit
monatlich
Format A4
2,80 DM

CPU
monatlich
Format A4
5,50 DM

Homecomputer
zweimonatlich
Format A4
6,– DM

Computronic
zweimonatlich
Format A4
6,50 DM

Spiele-Kassetten:
Commodore 64
VC-20, Atari,
TI-99, ZX-Spectrum,
Laser 2001,
Schneider CPC 464
Je 19,90 DM

D – 17/9

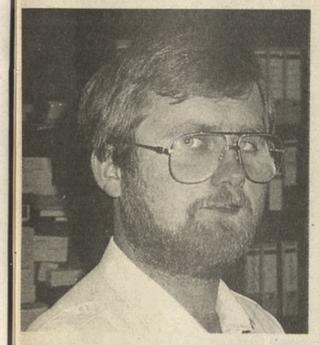
Spitzenprodukte

für den wachsenden Markt der Heimcomputer-Anwender

Informationen: Tronic-Verlag, Landstraße 29
Postfach 41, 3444 Wehretal 1, Ruf (0 56 51) 4 06 93
Vertrieb: Verlagsunion, Friedrich-Bergius-Straße 20, 6200 Wiesbaden, Ruf (0 61 21) 26 60

Liebe Leser der „CPU“!

Ab sofort erscheint „CPU“ als 2-monatige Doppelausgabe (Jetzt 92 Seiten!). Diese Entscheidung der CPU-Redaktion bringt Ihnen als Leser einen großen Vorteil: Durch den nun



größer gewordenen Vorbereitungszeitraum haben wir jetzt die Möglichkeit, aus einem umfangreicheren Programmangebot auswählen zu können, d. h. im Klartext: Die Qualität der veröffentlichten Programme wird verbessert. – Die Anzahl der veröffentlichten Programme wird erhöht. So warten schon in dieser Ausgabe 23 (!!) Programmbeiträge auf Sie!
Der Schwerpunkt der „Neuen CPU“ ist jetzt eindeutig unter dem Begriff „Tips und Tricks – Hilfen für die Arbeit mit Ihrem Computer“ festgelegt.
Hinzu kommt:

Ein neuer attraktiver Autorenwettbewerb (Insgesamt sind 600,– DM zu gewinnen!). Seite 82

Der Gewinner des Autorenwettbewerbs wird in der nächstfolgenden CPU-Ausgabe vorgestellt.

Wir werden auch in Zukunft die wichtigsten Computersysteme berücksichtigen! Wenn Sie Verbesserungsvorschläge haben oder Kritik äußern wollen, dann schreiben Sie uns.

Siegfried Görk, Chefredakteur

CPU – Der Inhalt:

USER — Programme für den Anwender

		Seite
TI-99:	Superfile	4
MSX:	Typer 60	6
MSX:	Disketten-Monitor	12
C-64:	Multi-Basic V3	19
C-64:	Bildeditor	23
CPC-464:	Machine Kit	29
CPC-464:	Basic + -Programm	36
Apple:	Digital Sound Editor (DSE)	40
VC-20:	Disassembler	45
Atari:	Error-Helper	47
ZX-Spectrum:	Basic-Erweiterung	50

UTILITY — Hilfsprogramme für den Programmierer

CPC-464:	Disc-Hilfe	52
CPC-464:	Druckart einstellen	53
ZX-Spectrum:	Trace und Single Step	54
ZX-Spectrum:	Variable List	56
VC-20	Old Programm	58
Atari:	Grafikausgabe des FX/ RX-80 Druckers	60
C-64:	Video-Utilities	63
Apple:	Basic-Befehle	65

HARDWARE

TI-99:	Eingaben-Modul mit oder ohne X-Basis	70
VC-20/C-64:	Bauanleitung für ein universelles Bussystem	72
ZX-Spectrum:	ISO-ROM	75

„Tips und Tricks“-Extra

TI-99:	Listschutz	75
TI-99:	Echtes Charset	75
CPC-464:	Wohin mit Maschinensprache?	77
VC-20:	Hilfen für den VC-20	78

Rubriken

Software-Katalog	90
Impressum	89
Infos	83
Computer-Club-Vorstellung	80
CPU-Podium	81

SUPERFILE FÜR DEN TI-99

Laden und Saven mit erhöhter Geschwindigkeit

Wohl jeder TI-Benutzer, der den Cassettenrecorder zur Datenspeicherung verwenden wollte, dürfte vom Computer zum Wahnsinn getrieben worden sein. Geschwindigkeit und Datensicherheit sind so schlecht, daß nur die Anschaffung einer Diskettenstation übrigblieb. Das Programm SUPERFILE macht diesen Problemen ein Ende, es speichert die Daten mit dem Vielfachen der normalen Geschwindigkeit und sogar noch erhöhter Datensicherheit ab. Sie benötigen dafür einen TI-99 mit Extended Basic und 32K-Speichererweiterung.

Zuerst ein kleiner Exkurs in das Betriebssystem des TI-99: Sollen Daten auf irgendeinem Peripheriegerät, sei es Diskette, Cassette oder Drucker, ausgegeben werden, so legt der Computer im Video-RAM einen Pufferspeicher an. In diesen werden solange Daten eingeschrieben, bis er voll ist. Erst dann werden die Informationen zum jeweiligen Peripheriegerät geschickt.

Beim Cassettenrecorder sieht das dann so aus: Sobald ein Datensatz übermittelt wird, geht er zunächst in den Pufferspeicher und sofort danach an den Cassettenrecorder. Bevor er jedoch gespeichert wird, wird ein Pfeifton aufgezeichnet, der zur Synchronisation beim erneuten Einlesen der Daten dient. Da nun jeder einzelne Datensatz, sei er ein Byte oder 255 Bytes lang, so übermittelt wird, addiert sich das bei größeren Datenfeldern zu unerträglichen Wartezeiten von bis zu 30 Minuten. Nicht zuletzt gibt es auch keine Möglichkeit, die übertragenen Daten zu prüfen (VERIFY).

Speichert man hingegen ein Basic-Programm auf Cassette ab, so geht das in erstaunlich kurzer Zeit und die Aufzeichnung läßt sich sofort danach auf Schreibfehler prüfen. Auch ist es auffällig, daß nur am Anfang der Aufzeichnung der Pfeifton gesendet wird – im Gegensatz zur Datenspeicherung, bei der dies ja vor jedem einzelnen Satz geschieht.

Die Lösung ist ganz einfach: Der Pufferbereich im Video-RAM ist bei der Programmaufzeichnung genauso lang wie das Programm – also maximal ca. 13 KByte, und nicht 255 Bytes wie sonst. Warum kann das Betriebssystem dann Daten nicht ebenfalls in einem Riesenpuffer unterbringen und dann auf einen Schlag speichern? Nun, in den 16 KByte des Video-RAMs belegen die Bildschirminformationen nur rund 3 KByte, der Rest wird vom Computer zur Stringspeicherung verwendet. Daher kommt es auch, daß nach dem SAVEn eines Programms plötzlich alle Variablen verschwunden sind. Sie werden sicherheitshalber vom Betriebssystem gelöscht, da nicht gewährleistet werden kann, daß sie nicht vom Programm-Puffer überschrieben wurden. Bei der Datenspeicherung hingegen müssen selbstverständlich alle Variablen erhalten werden, so daß irgendwo im Video-RAM 255 Bytes gesucht und die Daten von dort aus zur Cassette geschickt werden.

Das Programm SUPERFILE macht sich diese Besonderheiten des TI-Betriebssystems zunutze und speichert einen gesamten String-Array en bloc auf Cassette. Dazu ist es allerdings nötig, den Speicher mehrfach umzukopieren. Das dauert selbst in Maschinensprache noch einige Sekunden. Noch eine Vorbemerkung: Im High Memory müssen noch mindestens 8 KByte Speicherplatz frei sein. Dies können Sie mit dem SIZE-Befehl feststellen: Der Wert vor „BYTES OF PROGRAM SPACE FREE“ muß größer als 8000 sein. Außerdem kann der Low Memory nicht für weitere Maschinenprogramme benutzt werden.

Tippen Sie den Basic-Loader ab und speichern Sie diesen auf Kassette. Um damit arbeiten zu können, muß der Basic-Loader anschließend geladen und gestartet werden. Steht das Maschinenprogramm im Speicher, dann haben Sie zwei neue Befehle, mit denen Sie einen gesamten String-Array auf Cassette speichern und von dort wieder laden können.

Der erste Befehl heißt:

```
CALL LINK("CSAVE",Array$
      (),Count)
```

Für Array$() setzen Sie hier den Namen des gewünschten Datenfelds ein, für Count die Anzahl der zu speichernden Elemente des Stringarrays. Für Count = 150 werden dann die Feldelemente 0 bis 150 abgespeichert. Beachten Sie hier auch, daß nach dem Stringnamen leere Klammern folgen. Keinesfalls dürfen Sie in die Klammern eine Zahl oder Variable einsetzen, da dann unweigerlich eine Fehlermeldung ausgegeben wird.

Eingelesen werden die Daten mit:

```
CALL LINK("CLOAD",
      Array$())
```

Hier erkennt der Computer automatisch die Zahl der Datensätze. Die Namen der Datenfelder beim Speichern und Laden dürfen natürlich unterschiedlich sein. Haben Sie vergessen, vor dem Laden das Datenfeld mit DIM richtig zu dimensionieren, erscheint hier natürlich eine Fehlermeldung.

Alles klar? Hier noch zwei kurze Beispielprogramme:

```
100 ! SPEICHERN
110 DIM A$(200)
120 FOR I=1 TO 200
130 A$(I)="DIES IST DER
    DATENSATZ "&STR$(I)
140 NEXT I
150 CALL LINK("CSAVE",A$
    (),200)
160 END
```

Lassen Sie das Programm ablaufen, dann erscheinen nach kurzer Zeit die Meldungen zur Bedienung des Cassettenrecorders – sie funktionieren genauso wie beim Speichern eines Basic-Programms. Auch hier können Sie nach dem Speichern bei der Meldung "Check Tape (Y or N)?" entscheiden, ob die Daten geprüft werden sollen. Sie brauche wieder nur die Anweisungen des Computers zu befolgen. Sobald wieder READY erscheint, löschen Sie das Programm mit NEW und geben Sie das folgende Programm ein:

```
100 ! LADEN
110 DIM A$(200)
120 CALL LINK("CLOAD",
    A$())
130 FOR I=1 TO 200
140 PRINT A$(I)
150 NEXT I
160 END
```

Starten Sie das Programm und spulen Sie die Cassette an den Beginn der Aufzeichnung zurück. Nach kurzer Zeit erscheinen wieder die vorher gespeicherten Datensätze. Sie sehen das Maschinenprogramm ist sehr leicht zu bedienen und die neuen Befehle sind um vieles komfortabler als die Basic-Pendants. Dort müßten Sie erst die Datei im richtigen Format eröffnen, mit einer FOR-NEXT-Schleife die Daten schreiben, am Ende einen End-of-File-Text improvisieren und die Datei wieder schließen. Ähnlich umständlich ist das erneute Einlesen der Daten.

Nun für Maschinensprache-Freaks Erläuterungen zum Programm:

Das Problem bei der Programm-Entwicklung war, daß der Video-RAM zwar schon mit den benötigten Strings belegt ist, aber auch mit einer Menge anderer Daten: weiteren Strings, DEF-Definitionen, Unterprogrammlisten, Variablenlisten, Datei-Puffern usw. Man kann also nicht einfach den ganzen Video-RAM speichern und später wieder laden; das ladende Programm würde mit Sicherheit abstürzen, wenn plötzlich völlig andere Variablenlisten und Definitionen im Speicher stünden.

So sind mehrere Umkopiervorgänge in sämtlichen RAM-Bereichen notwendig:

SPEICHERN:

- 6 KBytes des High Memory in den freien Low Memory kopieren, die übrigen 8 KByte, die frei bleiben müssen, können hier nicht mehr gesichert werden.
- Die Variablen werden mit STRREF (String Reference) aus Basic geholt und in die insgesamt 14 (6+) KBytes des High Memory geschrieben.
- Jetzt wird der Inhalt des High Memory und des Video-RAMs byteweise ausgetauscht und die Daten vom Video-RAM aus auf Cassette geschrieben.
- Zum Schluß wird aus den 14 KByte des High Memory der frühere Inhalt des Video-RAMs wieder in den Video-RAM geschickt. Außerdem werden die im Low Memory gespeicherten 6 KBytes des High Memory dorthin zurückkopiert.

LADEN:

- Zuerst werden wieder 6 KByte des High Memory in den Low Memory kopiert, auch hier können die übrigen benötigten 8 KByte nicht gesichert werden.
- In einem Block wird der gesamte Video-RAM in den High Memory kopiert, worauf die Daten als Block von Cassette in den Video-RAM geholt werden.
- Nun wird Byte für Byte der Inhalt des mit den Informationen des Video-RAMs belegten High Memory und der Video-RAM ausgetauscht. Das heißt, daß jetzt der alte Inhalt des Video-RAMs wieder da ist und sämtliche eingelesenen Daten im High Memory stehen.
- Mit dem Hilfsprogramm STRASG (String Assignment) wird nun String für String wieder an den Basic-Interpreter übergeben. Zu guter Letzt wird der 6K-Bereich aus dem Low Memory wieder in den High Memory zurückgelegt.

Von diesen – zugegeben komplizierten – Vorgängen erfährt der Benutzer natürlich nichts, die beiden Unterprogramme CSAVE und CLOAD erledigen diese mühesame Arbeit automatisch für ihn – und zwar mit Hochgeschwindigkeit; die Beispielprogramm werden in rund 1 Minute 15 Sekunden bewältigt – gegenüber rund 20 Minuten im normalen Aufzeichnungsverfahren!

```
100 !  ********************
110 ! **********************
120 ! *                    *
130 ! *      SUPERFILE     *
140 ! *                    *
150 ! **********************
160 !  ********************
170 !
180 ! ----------------------
190 ! (C) Martin Kotulla 85
200 ! ----------------------
210 !
220 CALL CLEAR :: CALL HCHAR(1,5,42,25)
230 CALL CHAR(45,"00000000FF")
240 CALL CHAR(128,"DFBF60C0919B959191808080C060BFDF")
250 CALL CHAR(130,"FBFD06030101014951615149030 6FDFB")
260 CALL HCHAR(2,4,42,27):: CALL HCHAR(3,4,42):: CALL HCHAR(3,30,42)
270 DISPLAY AT(4,2):"*          SUPERFILE      *"
280 CALL MAGNIFY(3):: CALL SPRITE(#1,128,2,22,80)
290 CALL HCHAR(5,4,42):: CALL HCHAR(5,30,42):: CALL HCHAR(6,4,42,27)
300 CALL HCHAR(7,5,42,25)
310 CALL CHAR(91,"7E8199A1A199817E")
320 DISPLAY AT(10,5):"[ MARTIN KOTULLA 1985" :: CALL HCHAR(12,7,45,21)
330 DISPLAY AT(15,1):"LADEN:"
340 DISPLAY AT(17,1):"CALL LINK(""CLOAD"",A$())"
350 DISPLAY AT(20,1):"SPEICHERN:"
360 DISPLAY AT(22,1):"CALL LINK(""CSAVE"",A$(),NUM)"
370 CALL INIT
380 FOR I=9460 TO 10034 STEP 4 :: READ A,B,C,D :: CALL LOAD(I,A,B,C,D)
390 NEXT I :: CALL LOAD(8194,63,255,63,240)
400 CALL LOAD(16368,67,76,79,65,68,32,37,50,67,83,65,86,69,32,37,154)
410 CALL CLEAR :: END
420 !
430 DATA 255,000,000,000,000,003,004,000,008,000,000,058,000,255,006,012
440 DATA 009,196,080,080,016,000,096,003,067,083,049,000,255,255,205,096
450 DATA 033,131,035,253,038,184,040,183,041,182,042,195,043,193,044,179
```

```
460 DATA 045,194,047,196,058,181,059,180,060,191,061,190,062,192,002,224
470 DATA 037,018,002,006,054,059,200,006,037,008,006,160,038,062,002,000
480 DATA 009,196,002,001,160,000,002,002,054,059,004,032,032,044,002,000
490 DATA 003,208,002,001,005,000,004,032,032,032,216,032,036,246,131,124
500 DATA 004,032,038,186,000,061,006,160,038,152,004,192,002,001,000,001
510 DATA 002,002,160,000,209,018,009,132,004,032,032,016,160,132,005,130
520 DATA 136,018,037,000,019,006,136,034,000,002,037,000,019,002,005,128
530 DATA 016,241,004,096,038,030,002,224,037,018,004,192,002,001,000,002
540 DATA 004,032,032,012,004,032,032,024,018,184,193,160,131,074,006,160
550 DATA 038,062,004,192,002,001,000,001,002,002,160,000,004,199,212,160
560 DATA 036,244,004,032,032,020,208,210,009,131,161,195,005,135,160,131
570 DATA 005,130,005,128,129,128,018,243,002,039,000,004,005,130,196,160
580 DATA 037,000,002,004,003,214,192,135,192,071,006,160,035,230,006,160
590 DATA 038,152,002,000,003,208,002,001,006,000,004,032,032,032,216,032
600 DATA 036,246,131,124,004,032,038,186,000,061,002,000,009,196,002,001
610 DATA 160,000,002,002,054,059,004,032,032,036,002,000,039,116,002,001
620 DATA 190,002,002,002,012,028,204,112,006,002,024,253,004,032,038,186
630 DATA 000,052,002,224,131,224,004,096,000,112,002,000,003,208,002,001
640 DATA 037,002,002,002,000,018,004,032,032,036,002,008,191,064,136,008
650 DATA 131,134,026,004,002,000,011,000,004,032,032,052,200,032,037,012
660 DATA 131,074,216,032,037,014,131,076,002,000,003,221,200,000,131,086
670 DATA 200,032,036,248,131,084,004,224,131,208,216,032,036,252,131,109
680 DATA 002,000,039,116,002,001,190,002,002,002,012,028,204,049,006,002
690 DATA 024,253,004,091,002,000,009,196,002,003,160,000,002,005,054,059
700 DATA 004,032,032,040,193,001,208,083,004,032,032,032,220,196,005,128
710 DATA 006,005,022,246,004,091,037,018,038,190,216,032,152,002,063,222
720 DATA 216,032,152,002,063,223,006,032,063,222,216,062,156,002,216,062
730 DATA 156,002,002,224,131,224,002,000,131,230,002,001,063,182,002,002
740 DATA 000,013,204,112,006,002,022,253,184,032,036,250,131,115,208,160
750 DATA 131,115,009,130,002,034,130,254,002,000,039,012,006,064,204,128
760 DATA 196,160,036,254,004,096,000,112,006,096,131,115,216,032,063,222
770 DATA 156,002,216,032,063,223,156,002,002,000,131,230,002,001,063,182
780 DATA 002,002,000,013,204,049,006,002,022,253,002,224,037,018,003,128
790 !
800 ! PRUEFSUMME:
810 !
820 RESTORE
830 FOR I=0 TO 572 STEP 4 :: READ A,B,C,D :: SUM=SUM+A+B+C+D
840 NEXT I :: DISPLAY BEEP:"DIE DATA-ZEILEN SIND"
850 IF SUM=41849 THEN PRINT "KORREKT!" ELSE PRINT "FEHLERHAFT!"
860 END
```

MSX-Programm: „TYPER-60“

MSX Computer benutzen den TMS 9929 A Videoprozessor für die Bildaufbereitung. Dieser Videoprozessor arbeitet mit maximal 40 Zeichen pro Zeile, was eine einigermaßen vernünftige Textverarbeitung ausschließt, denn mit 40 Zeichen pro Zeile läßt sich in dieser Hinsicht nicht viel anfangen.
Da die MSX Computer jedoch an jeden handelsüblichen Fernseher angeschlossen werden können, wäre ein 80 Zeichen Videoprozessor ungeeignet, weil ein Fernseher 80 Zeichen fast nicht mehr lesbar darstellen kann.
Bei der Textverarbeitung wird jedoch meistens DIN A4 Papier benutzt, worauf 80 Zeichen je Zeile passen. Jedoch wird das Papier nicht von Rand zu Rand beschrieben, sodaß man auch mit weniger als 80 Zeichen arbeiten könnte.

TYPER-60 schafft es jetzt bei allen MSX Computern 60 Zeichen pro Zeile darzustellen, ohne daß hierzu eine zusätzliche Hardwareerweiterung erforderlich ist.
Wie das geht? Ganz einfach, indem man den hochauflösenden Grafikmodus als Textmodus „entfremdet“.
Der hochauflösende Grafikmodus generiert das Bild in 256 mal 192 Punkten, was bei einer 8x8 Matrix 32 mal 24 Zeichen entsprechen würde.
Jetzt wird ein Zeichensatz generiert, welcher die Zeichen in eine 4x8 Matrix auflöst. Beim Schreiben von Texten wird nun je nach Cursorposition abwechselnd in die linke oder rechte Hälfte einer 8x8 Zeichenposition das entsprechend aufgenerierte Zeichen gepoked. Indem pro Zeichenposition immer Zeichen dargestellt werden kann man 64 Zeichen pro Zeile ausdrucken. TYPER-60 benutzt jetzt von diesen 64 Zeichen lediglich 60 Zeichen, da bei den meisten Fernsehgeräten die rechte und linke Randspalte nicht mehr dargestellt wird, und somit jeweils die beiden rechten und linken Zeichen nicht mehr sichtbar sein würden.

Nach dem Programmstart und der Bildaufgenerierung meldet TYPER-60 mit einem Piepston seine Einsatzbereitschaft. In der untersten Programmzeile steht Ihnen jetzt eine Auswahl von 5 Funktionen der Funktionstasten zur Verfügung.
„type print save load clear“ diese haben folgende Bedeutung:

<F1>-type- Der gerade angezeigte Text kann bearbeitet werden, oder ein neuer Text wird geschrieben, wenn sich kein Text im Speicher befindet.

<F2>-print- Der Text im Speicher wird ausgedruckt.

<F3>-save- Der Text wird gesichert. Hierzu fragt der Rechner nach dem FILE NAME, unter welchem Sie den Text speichern wollen.

<F4>-load- Ein Text wird geladen. Auch hier erhalten Sie die Abfrage nach dem „File Name“, der in der gleichen Form wie bei der Option -save- einzugeben ist.

<F5>-clear- Diese Option löscht den Textspeicher.

TYPER-60 teilt ein Blatt in 3 Sektoren, wobei jeder Sektor 22 Zeilen aufnehmen kann.

Der Sektor wird immer in der oberen Bildschirmzeile eingeblendet.

Sektor 1 enthält die Zeilen 1 bis 22, Sektor 2 die Zeilen 23 bis 44 und Sektor 3 die Zeilen 45 bis 66.

Die Optionen -load- und -clear- gehen automatisch nach der Ausführung in Sektor 1.

Die Option -type- hingegen springt in den gerade angezeigten Sektor.

In den Optionen -load- und -save- wird mittels der <BS> Taste die Eingabe gelöscht und <ESC> springt zum Hauptmenü. Sobald Sie sich im Schreibmodus befinden, wird in der oberen linken Bildschirmecke ein rotes Quadrat sichtbar. Dies ist Ihr Cursor, der die aktuelle Schreibposition angibt. Sie können jetzt den Cursor beliebig mit den Cursortasten auf dem Bildschirm bewegen.

Sobald Sie den Bildschirm mit dem Cursor nach unten verlassen, wechselt der Bildschirm in den nächsthöheren Sektor (max. Sektor 3).

Verlassen Sie den Bildschirm nach oben, so wird in den nächst niedrigeren Sektor gewechselt (min. Sektor 1).

Durch die Aufbereitung von 60 Zeichen pro Zeile wird die Zeichendarstellung gegenüber der Schreibgeschwindigkeit etwas verzögert. TYPER-60 speichert jedoch alle eingegebenen Zeichen, und holt die Darstellung nach.

Um deutsche Umlaute darzustellen, muß Ihr Drucker auf den deutschen Zeichensatz geschaltet werden. Sehen Sie in Ihrem Druckerhandbuch nach und schalten Sie den Drucker entsprechend um.

Zur Eingabe werden jetzt nicht die Umlauttasten benutzt, sondern die, dem Druckerzeichensatz zugeordneten Tasten. Die Darstellung auf dem Bildschirm erfolgt als Umlaut. Sollte Ihr Drucker nicht auf deutschen Zeichensatz umschaltbar sein, so dürfen keine Umlaute benutzt werden.

Sektorenwechsel:
Der Sektor wird automatisch gewechselt, sobald der Cursor den Bildschirm nach oben oder unten verläßt. Sie haben jedoch die Möglichkeit einen Sektor direkt anzuspringen, indem Sie eine der Funktionstasten <F1> bis <F3> drücken. <F1> springt in Sektor 1, <F2> in Sektor 2 und <F3> springt in Sektor 3.

Bei einem Sektorenwechsel wird zuerst der Bildschirm gelöscht und danach der Text auf den Bildschirm geschrieben. Auch während dieser Zeit werden Anschläge gespeichert und danach nachgeholt.

Die Taste <HOME> springt automatisch in Sektor 1 zurück. Während der Schriftausgabe kann bereits der Sektor wieder gewechselt werden. Dabei wird die Schriftausgabe abgebrochen und der neu angewählte Sektor wird ausgegeben.

Um aus dem Schreibmodus in das Hauptmenü zu kommen, drücken Sie die <ESC> Taste.

Hinweise für Benutzer der Quick Disk

TYPER-60 kann in der abgedruckten Form nur Cassette oder Diskette zur Speicherung von Texten ansprechen.

Besitzer einer Quick Disk müssen die Programmzeilen 880 und 950 wie folgt ändern, um auch die Quick Disk ansprechen zu können:

```
880 IF LEFT$(AF,2)="QD" THEN CALL BSAVE(AF,52001!,55961!) ELSE BSAVE AF, 52001!,55961!
950 IF LEFT$(AF,2)="QD" THEN CALL BLOAD(AF) ELSE BLOAD AF
```

Jetzt kann auch die Quick Disk zur Textspeicherung angesprochen werden.

Für Disketten oder Quick Disk Betrieb gibt eine wesentliche schnellere Version von TYPER-60. Benutzer von Diskette oder Quick Disk können sich gerne deshalb mit dem Programmautor in Verbindung setzen.

Editieren im Text:
Zeilenorientierter Editor:

<INS> Taste setzt links der Cursorposition ein Leerzeichen ein. Das letzte Zeichen der Zeile geht dabei verloren.

<DEL> Taste löscht das Zeichen der Cursorposition und rückt den Text rechts des Cursors auf. Als letztes Zeichen wird der Zeile ein Leerzeichen angefügt.

<BS> Taste löscht das Zeichen links der Cursorposition auf. Als letztes Zeichen wird der Zeile ein Leerzeichen angefügt.

Die Ausführung dieser Operationen dauern ca. 4 Sekunden. Während dieser Zeit kann jedoch weitergeschrieben werden, da alle geschriebenen Zeichen nachgeholt werden.

Seitenorientierter Editor:

<F4> -ins- setzt eine Leerzeile vor der Zeile der Cursorposition ein.

<F5> -del- löscht die der Cursorposition entsprechende Zeile und rückt den nachfolgenden Text auf.

Bei diesen Operationen wird der Indikator ”WAIT“ angezeigt.

Die Zeit für das Einsetzen und Löschen von Zeilen ist abhängig von der Zeilenposition.

Fortsetzung S. 8

Wird eine Zeile ziemlich weit oben im Text gelöscht oder eingesetzt, so dauert dies natürlich länger, als wenn die Operation weiter unten im Text durchgeführt wird.

Sie können jedoch auch hier weiterschreiben, und die Zeichen werden nachgeholt.

Was tun bei einem Fehler!
Sollte TYPER-60 durch einen Dateifehler (Diskette voll, Textdatei nicht gefunden usw.) abbrechen, so drücken Sie die Funktionstaste <F5>. Der Bildschirm wird gelöscht, und Sie haben wieder die Hauptauswahl:
„type print save load clear"

Gehen Sie mit <F1>-type- in den Schreibmodus. Drücken Sie sobald der Cursor erscheint die Funktionstaste <F1> für Sektor 1, und Ihr bereits geschriebener Text erscheint wieder.

Belegung der Tasten für deutsche Umlaute

Taste	Umlaut
[	Ä
\	Ö
]	Ü
(	ä
i	ö
)	ü
~	ß
Ω	§

„Typer 60": Bei allen MSX-Computern – Darstellung von 60 Zeichen pro Zeile

```
10 '┌──────────────────────────────┐
15 '|        T Y P E R    6 0     |
20 '├──────────────────────────────┤
25 '|   Textverarbeitung für MSX   |
30 '|   mit 60 Zeichen pro Zeile   |
35 '|    Bildschirmaufbereitung    |
40 '├──────────────────────────────┤
45 '|   (c)1985 by Volker Becker   |
50 '|      Steinbacher Str.10      |
55 '|       6370 Oberursel 6       |
60 '└──────────────────────────────┘
65 '
70 '
75 CLEAR 200,52000!
80 DEFINT I,J,K,N
85 DEFSTR A,B,C
90 DIM NF(2)
95 KEY 5,"GOTO 110"+CHR$(13):KEY OFF:COLOR 15,4,4:CLS:PRINT"T Y P E R - 6 0":PRI
NT"---------------"
100 LOCATE 0,3:PRINT"Textspeicher wird gelöscht"
105 FOR I=1 TO 66:FOR K=1 TO 60:POKE 52000!+60*(I-1)+K,32:NEXT K,I:LZ=0
110 CLOSE:COLOR 9,12,12:SCREEN 2,0,1:GOSUB 1070
115 FOR I=&H2000 TO &H2000+255:VPOKE I,248:NEXT:FOR I=&H2000+256 TO &H36FF:VPOKE
 I,244:NEXT:FOR I=&H3700 TO &H37FF:VPOKE I,248:NEXT
120 OPEN "GRP:" FOR OUTPUT AS #1
125 FOR I=1 TO 5:BEEP:NEXT
130 COLOR 15,8:PRESET(16,0):COLOR 15:PRINT #1,"TYPER 60 (c)1985 by V.Becker"
135 GOTO 720
140 IP=1:IZ=-1:JZ=1:JS=1:NS=0
145 PUT SPRITE 0,(4*(JS),8*(JZ)-1),9,128
150 AV=INKEY$:IF AV="" THEN 150
155 IF AV=CHR$(13) THEN JS=1:IP=1:IZ=-1:GOTO 425
160 IF AV=CHR$(8) THEN 270
165 IF AV=CHR$(30) THEN 450
170 IF AV=CHR$(31) THEN 425
175 IF AV=CHR$(29) THEN 355
180 IF AV=CHR$(28) THEN 380
185 IF AV=CHR$(11) AND NS>0 THEN NS=0:GOSUB 510:IP=1:JS=1:JZ=1:IZ=-1:GOTO 225
190 IF AV=CHR$(127) THEN 315
195 IF AV=CHR$(18) THEN 335
200 IF AV=CHR$(27) THEN 720
205 IA=ASC(AV):IF IA<32 OR IA>126 THEN 150
210 IF JS=52 THEN PLAY "t80s9m10000o7o2"
215 IF JS>60 THEN GOTO 150
220 ON IZ+2 GOSUB 240,255:IZ=NOT IZ:POKE 52000!+60*(22*NS+(JZ-1))+JS,IA:JS=JS+1
225 IF JS>60 THEN 150
230 PUT SPRITE 0,(4*(JS),8*(JZ)-1),9,128
235 GOTO 150
240 FOR I=0 TO 7:VPOKE 256*JZ+8*IP+I,16*VPEEK(8*IA+&H3800+I)+(VPEEK(256*JZ+8*IP+
I)AND 15):NEXT
245 IF LZ<22*NS+JZ THEN LZ=22*NS+JZ
250 RETURN
255 FOR I=0 TO 7:VPOKE 256*JZ+8*IP+I,VPEEK(8*IA+&H3800+I)+(VPEEK(256*JZ+8*IP+I)A
ND 240):NEXT:IP=IP+1
260 IF LZ<22*NS+JZ THEN LZ=22*NS+JZ
265 RETURN
270 IF JS=1 THEN 225 ELSE JS=JS-1:GOSUB 275:GOTO 225
275 PUT SPRITE 0,(4*(JS),8*(JZ)-1),9,129
280 FOR I=JS TO 59:POKE 52000!+60*(22*NS+(JZ-1))+I,PEEK(52000!+60*(22*NS+(JZ-1))
+I+1):NEXT:POKE 52000!+60*(22*NS+(JZ-1))+60,32
285 I1=IP:I2=IZ
290 IP=1:IZ=-1
295 N1=1
300 GOSUB 405
305 IZ=NOT I2:IF NOT IZ THEN IP=I1-1 ELSE IP=I1
310 VPOKE &H1B02,128:RETURN
315 PUT SPRITE 0,(4*(JS),8*(JZ)-1),9,129
320 FOR I=JS TO 59:POKE 52000!+60*(22*NS+(JZ-1))+I,PEEK(52000!+60*(22*NS+(JZ-1))
+I+1):NEXT:POKE 52000!+60*(22*NS+(JZ-1))+60,32
325 N1=1:GOSUB 405
330 VPOKE &H1B02,128:GOTO 225
335 PUT SPRITE 0,(4*(JS),8*(JZ)-1),9,129
340 FOR I=60 TO JS+1 STEP -1:POKE 52000!+60*(22*NS+(JZ-1))+I,PEEK(52000!+60*(22*
NS+(JZ-1))+I-1):NEXT:POKE 52000!+60*(22*NS+(JZ-1))+JS,32
345 N1=1:GOSUB 405
350 VPOKE &H1B02,128:GOTO 225
355 IF JS=1 THEN 225 ELSE JS=JS-1
360 I1=IP:I2=IZ
365 IP=1:IZ=-1
370 IZ=NOT I2:IF NOT IZ THEN IP=I1-1 ELSE IP=I1
375 GOTO 225
380 IF JS>=60 THEN 225 ELSE JS=JS+1
385 I1=IP:I2=IZ
390 IP=1:IZ=-1
395 IZ=NOT I2:IF IZ THEN IP=I1+1 ELSE IP=I1
400 GOTO 225
405 FOR K=60*(22*NS+(JZ-1))+1 TO 60*(22*NS+(JZ-1))+59 STEP 2:IA=PEEK(52000!+K):I
B=PEEK(52000!+K+1)
410 IF IA=32 AND IB=32 THEN FOR I=0 TO 7:VPOKE 256*JZ+8*N1+I,0:NEXT:GOTO 420
415 FOR I=0 TO 7:VPOKE 256*JZ+8*N1+I,16*VPEEK(&H3800+8*IA+I)+VPEEK(&H3800+8*IB+I
):NEXT
420 N1=N1+1:NEXT:RETURN
425 JZ=JZ+1
430 IF JZ>22 THEN NS=NS+1 ELSE 225
435 IF NS>2 THEN NS=2:JZ=22:GOTO 225
440 GOSUB 510
445 JZ=1:JS=1:IP=1:IZ=-1:GOTO 225
450 JZ=JZ-1
455 IF JZ<1 THEN NS=NS-1 ELSE 225
460 IF NS<0 THEN NS=0:JZ=1:GOTO 225
465 GOSUB 510
470 JZ=22:JS=1:IP=1:IZ=-1:GOTO 225
475 VPOKE &H1B02,200:GOSUB 530:VPOKE &H1B02,129:FOR JZ=1 TO 22:N1=1:VPOKE &H1B00
,JZ*8:IF (22*NS+JZ)>LZ THEN RETURN
```

```
480 FOR K=60*(22*NS+(JZ-1))+1 TO 60*(22*NS+(JZ-1))+59 STEP 2:IA=PEEK(52000!+K):I
B=PEEK(52000!+K+1):VPOKE &H1B01,8*N1
485 IF IA=32 AND IB=32 THEN GOTO 495
490 FOR I=0 TO 7:VPOKE 256*JZ+8*N1+I,16*VPEEK(&H3800+8*IA+I)+VPEEK(&H3800+8*IB+I
):NEXT
495 N1=N1+1:NEXT
500 NEXT JZ:RETURN
505 VPOKE &H1B02,200:GOSUB 530:RETURN
510 FOR J=0 TO 2:IF (22*NS)<LZ THEN NF(J)=-1 ELSE NF(J)=0
515 NEXT
520 ON NF(NS)+2 GOSUB 475,505
525 RETURN
530 FOR I=1 TO 5:KEY(I) OFF:NEXT
535 FOR I=&H1700 TO &H17FF:VPOKE I,0:NEXT
540 AX="WAIT":GOSUB 575:FOR I=0 TO &H16FF:VPOKE I,0:NEXT:GOSUB 580
545 COLOR 15,8:PRESET(16,0):COLOR 15:PRINT #1,"TYPER 60              Sektor";NS+1
550 LINE (0,7)-(255,7),1
555 COLOR 15,8:PRESET(16,185):PRINT #1,"Sekt.1 Sekt.2 Sekt.3 ins del"
560 LINE (0,184)-(255,184),1
565 ON KEY GOSUB 585,590,595,600,660
570 FOR I=1 TO 5:KEY(I) ON:NEXT:RETURN
575 FOR I=1 TO 5:KEY(I) OFF:NEXT:PA=116:FOR I=1 TO 4:PUT SPRITE I,(PA,255),15,AS
C(MID$(AX,I,1)):PA=PA+4:NEXT:RETURN
580 FOR I=1 TO 5:KEY(I) ON:NEXT:FOR I=1 TO 4:PUT SPRITE I,(0,0),0,0:NEXT:RETURN
585 NS=0:GOSUB 510:IP=1:JS=1:JZ=1:IZ=-1:RETURN 225
590 NS=1:GOSUB 510:IP=1:JS=1:JZ=1:IZ=-1:RETURN 225
595 NS=2:GOSUB 510:IP=1:JS=1:JZ=1:IZ=-1:RETURN 225
600 AX="WAIT":GOSUB 575:VPOKE &H1B03,0
605 IF (22*NS+JZ)>LZ THEN KY=JZ:GOTO 655
610 FOR II=LZ TO 22*NS+JZ STEP-1:N1=1
615 FOR K=1 TO 60:POKE 52000!+60*((II))+K,PEEK(52000!+60*((II-1))+K):NEXT K,II
620 FOR K=1 TO 60:POKE 52000!+60*(22*NS+(II))+K,32:NEXT K
625 LZ=LZ+1:IF LZ>66 THEN LZ=66
630 KY=JZ:FOR JZ=KY TO 22:N1=1:IF (22*NS+JZ)>LZ THEN 655
635 FOR K=60*(22*NS+(JZ-1))+1 TO 60*(22*NS+(JZ-1))+59 STEP 2:IA=PEEK(52000!+K):I
B=PEEK(52000!+K+1):VPOKE &H1B01,8*N1
640 FOR I=0 TO 7:VPOKE 256*JZ+8*N1+I,16*VPEEK(&H3800+8*IA+I)+VPEEK(&H3800+8*IB+I
):NEXT
645 N1=N1+1:NEXT
650 NEXT JZ
655 JZ=KY:JS=1:IP=1:IZ=-1:GOSUB 580:RETURN 225
660 AX="WAIT":GOSUB 575:VPOKE &H1B03,0
665 IF (22*NS+JZ)>LZ THEN KY=JZ:GOTO 715
670 FOR II=22*NS+JZ TO LZ:N1=1
675 FOR K=1 TO 60:POKE 52000!+60*((II-1))+K,PEEK(52000!+60*((II))+K):NEXT K,II
680 FOR K=1 TO 60:POKE 52000!+60*(22*NS+(II))+K,32:NEXT K
685 LZ=LZ-1:IF LZ<1 THEN LZ=1
690 KY=JZ:FOR JZ=KY TO 22:N1=1:IF (22*NS+JZ)>(LZ+1) THEN 715
695 FOR K=60*(22*NS+(JZ-1))+1 TO 60*(22*NS+(JZ-1))+59 STEP 2:IA=PEEK(52000!+K):I
B=PEEK(52000!+K+1):VPOKE &H1B01,8*N1
700 FOR I=0 TO 7:VPOKE 256*JZ+8*N1+I,16*VPEEK(&H3800+8*IA+I)+VPEEK(&H3800+8*IB+I
):NEXT
705 N1=N1+1:NEXT
710 NEXT JZ
715 JZ=KY:JS=1:IP=1:IZ=-1:GOSUB 580:RETURN 225
720 GOSUB 580:VPOKE &H1B03,0:FOR I=1 TO 5:KEY(I) OFF:NEXT
725 FOR I=&H1700 TO &H17FF:VPOKE I,0:NEXT
730 ON KEY GOSUB 750,775,820,895,965
735 COLOR 15,8:PRESET (16,185):PRINT #1,"type  print save  load clear"
740 FOR I=1 TO 5:KEY(I) ON:NEXT
```



```
745 GOTO 745
750 FOR I=1 TO 5:KEY(I) OFF:NEXT
755 GOSUB 1000
760 FOR I=&H1700 TO &H17FF:VPOKE I,0:NEXT
765 IP=1:IZ=-1:JZ=1:JS=1
770 GOSUB 545:RETURN 145
775 FOR I=1 TO 5:KEY(I) OFF:NEXT
780 GOSUB 1000
785 FOR I=&H1700 TO &H17FF:VPOKE I,0:NEXT
790 AX="prnt":GOSUB 575
795 COLOR 15,8:PRESET (16,185):PRINT #1,"Druckvorgang ist eingeleitet"
800 FOR I=1 TO LZ
805 P$="":FOR K=1 TO 60:P$=P$+CHR$(PEEK (52000!+60*(I-1)+K)):NEXT
810 LPRINT STRING$(10,32)+P$:NEXT
815 RETURN 720
820 FOR I=1 TO 5:KEY(I) OFF:NEXT
825 GOSUB 1000
830 FOR I=&H1700 TO &H17FF:VPOKE I,0:NEXT
835 AX="save":GOSUB 575
840 COLOR 15,8:PRESET (16,185):PRINT #1,"File Name: ":AF=""
845 AV=INKEY$:IF AV="" THEN 845
850 IF AV=CHR$(8) THEN 830
855 IF AV=CHR$(13) THEN 870
860 IF AV=CHR$(27) THEN RETURN 720
865 AF=AF+AV:PRESET (112,185):PRINT #1,AF:GOTO 845
870 CLOSE:POKE 55961!,LZ
875 AX="WAIT":GOSUB 575
880 BSAVE AF,52001!,55961!
885 GOSUB 580:OPEN "GRP:" FOR OUTPUT AS #1
890 RETURN 720
895 FOR I=1 TO 5:KEY(I) OFF:NEXT
900 GOSUB 1000
905 FOR I=&H1700 TO &H17FF:VPOKE I,0:NEXT
910 AX="load":GOSUB 575
915 COLOR 15,8:PRESET (16,185):PRINT #1,"File Name: ":AF=""
920 AV=INKEY$:IF AV="" THEN 920
925 IF AV=CHR$(8) THEN 905
930 IF AV=CHR$(13) THEN 945
935 IF AV=CHR$(27) THEN RETURN 720
940 AF=AF+AV:PRESET (112,185):PRINT #1,AF:GOTO 920
945 AX="WAIT":GOSUB 575:CLOSE
950 BLOAD AF
955 GOSUB 580:OPEN "GRP:" FOR OUTPUT AS #1:LZ=PEEK(55961!)
960 VPOKE &H1B03,9:GOTO 585
965 FOR I=1 TO 5:KEY(I) OFF:NEXT
970 GOSUB 1000
975 FOR I=&H1700 TO &H17FF:VPOKE I,0:NEXT
980 AX="CLR ":GOSUB 575
985 COLOR 15,8:PRESET (16,185):PRINT #1," Textspeicher wird gelöscht"
990 FOR I=1 TO 66:FOR K=1 TO 60:POKE 52000!+60*(I-1)+K,32:NEXT K,I
995 NS=0:GOSUB 530:LZ=0:RETURN 140
1000 FOR I=0 TO &HFF:VPOKE I,0:NEXT:RETURN
1005 DATA 0000000000000000,0404040404000400,0A0A0A0000000000,00000A0E0A0E0A00,04
0E0C04060E0400,020A0C04060A0800,040A04080A040200,0202040000000000
1010 DATA 0408080808080400,0402020202020400,0000040E040E0400,000004040E040400,00
0000000404040800,000000000E000000,0000000000C0C00,0202040404080800
1015 DATA 040A0A0E0A0A0400,040C0404040E0E00,0E02020E08080E00,0C02020C02020C00,08
080A0E02020200,0E08080C02020C00,0408080C0A0A0400,0E02020404040400
1020 DATA 040A0A040A0A0400,040A0A0602020400,0000040000040000,0000040000040408,00
0002040804020 0,0000000E000E0000,0000080402040800,040A020404000400
1025 DATA 040A0C0A0A060A04,040A0A0A0E0A0A00,0C0A0A0C0A0A0C00,0608080808080600,0C
```

```
0A0A0A0A0A0C00,0E08080C08080E00,0E08080C08080800,0E08080E0A0A0E00
1030 DATA 0A0A0A0E0A0A0A00,0404040404040400,0202020200A0A400,0A0A0A0C0A0A0A00,08
0808080808080E00,0A0E0A0A0A0A0A00,0A0A0C0A060A0A00,040A0A0A0A0A0400
1035 DATA 0C0A0A0C08080800,040A0A0A0A0A0E0600,0C0A0A0C0A0A0A00,0E08080E02020E00,0E
0404040404040400,0A0A0A0A0A0A0E00,0A0A0A0A0A040404,0A0A0A0A0A0A0E0A00
1040 DATA 0A0A0A040A0A0A00,0A0A0A0A0404040400,0E0202040808080E00,0A040A0A0E0A0A00,0A
040A0A0A0A0A0400,0A000A0A0A0A0A0E00,040A000000000000,0000000000000F00
1045 DATA 0402000000000000,00000E020E0A0E00,08080C0A0A0A0C00,0000060808080600,02
02060A0A0A0A0600,00000E0A0E080E00,040A080C08080800,00000E0A0A0E020E
1050 DATA 08080C0A0A0A0A00,0400040404040400,0200020200A0A400,0808080A0A0C0A00,0C
0404040404040E00,00000A0E0E0A0A00,00000C0A0A0A0A00,0000040A0A0A0400
1055 DATA 00000C0A0A0A0C08,0000060A0A0A0602,00000A0C08080800,00000E080E020E00,04
040E0404040600,00000A0A0A0A0E00,00000A0A0A040400,00000A0A0A0E0A00
1060 DATA 00000A0A040A0A00,00000A0A0A0E020E,00000E0204080E00,0A000E020E0A0E00,0A
00040A0A0A0400,0A000A0A0A0A0400,040A0C0A0A0A0C08,00002050F8000000
1065 DATA 1F1111111111111F,000000000011111F
1070 RESTORE 1005
1075 FOR I=32 TO 129:READ C:FOR K=0 TO 7:VPOKE 8*I+K+&H3800,VAL("&H0"+MID$(C,2*K
+1,2)):NEXT K,I:RETURN
```

Disketten-Monitor für MSX-Computer

MSX-Disketten sind 100% IBM-kompatibel. Jetzt gibt es jedoch Funktionen, die unter IBM DOS 2.0 durchführbar sind, zwar von MSX DISK BASIC und MSX DOS nicht angesprochen, aber erkannt werden. MSX DISK BASIC ist jedoch gegenüber dem IBM DISK BASIC um einige Funktionen erweitert, die das direkte Lesen und Schreiben von Diskettensektoren erlauben.

DISKETTEN-MONITOR erlaubt es nun, nicht nur diese verborgenen Funktionen auf den MSX-Computern nutzbar zu machen, sondern geht noch darüber hinaus. So ist es z. B. möglich, einer Diskette einen Diskettennamen zu „verpassen", auch wenn diese bereits formatiert ist. Ferner kann dieser Name jederzeit nachträglich vergeben und auch geändert werden.

Weiterhin kann man Dateien derart schützen, daß diese nicht mehr im Inhaltsverzeichnis erscheinen und auch nicht mehr geladen werden können. Die Dateien können dann aber auch nicht mehr versehentlich überschrieben werden.

Nach der Aufhebung eines solchen Schutzes kann die Datei wieder wie gewohnt verarbeitet werden.

Des weiteren ist ein direkter Zugriff auf die Sektoren der Diskette möglich. Hier kann direkt auf der Diskette editiert werden. DISKETTEN-MONITOR bleibt nichts auf der Diskette geheim, denn er erkennt die Formatierungsart der Diskette (es gibt 8 verschiedene Arten) und sogar, welches System bzw. welches Laufwerk die Diskette formatiert hat.

Bei all diesen Operationen bleiben die Disketten 100% IBM-kompatibel. Wenn Sie z. B. einer Diskette mit dem DISKETTEN-MONITOR einen Namen vergeben, so erkennt auch der IBM PC diesen Namen und umgekehrt. Geschützte Dateien bleiben auch beim IBM PC verborgen. DISKETTEN-MONITOR gibt Ihnen jedoch auch geschützte Dateien in seinem internen Inhaltsverzeichnis aus, sodaß Sie jederzeit einen Gesamtüberblick über den Diskinhalt haben.

Tippen Sie das Programm ein, und sichern Sie dieses auf 2 verschiedene Disketten ab, bevor Sie das Programm zum ersten Mal starten. Dies ist unbedingt notwendig, da durch einen Tippfehler bei Sektorenzugriffen die ganze Diskette unbrauchbar werden könnte und dann neu formatiert werden muß.

Starten Sie das Programm jetzt mit RUN. Geben Sie das aktuelle Datum im Format TT.MM.JJJJ ein, wobei das Jahr immer 4stellig (z. B. 1985 anstatt 85) eingegeben werden muß. Danach geben Sie noch das aktuelle Laufwerk (A oder B) ein, anschließend erhalten Sie das Menü als Funktionstasteneinblendung:

<F 1>-Status Diese Auswahl gibt den Status der Diskette auf dem Monitor aus. Es wird angezeigt, welches System die Diskette formatiert hat, wie viele Spuren mit wie vielen Sektoren bestehen, ob die Diskette einseitig oder zweiseitig formatiert ist, wieviel Speicherplatz vom System, geschützten - und Normaldateien belegt ist und wieviel Speicherplatz noch verfügbar ist. Ferner wird noch der Diskettenname ausgegeben.

<F 2>-Name Mit dieser Auswahl kann der Diskettenname geändert werden. Es wird zuerst der derzeitige Name angezeigt und danach nach dem neuen Namen gefragt. Wird kein neuer Name eingegeben, so bleibt der alte Name unverändert. Wenn die Diskette noch keinen Namen hat, so wird mit dieser Auswahl ein Name vergeben.

<F 3>-Inhalt Diese Auswahl gibt das Inhaltsverzeichnis der Diskette auf dem Monitor aus.
Es wird der Diskettenname angezeigt. Anschließend werden alle auf der Diskette befindlichen Dateien mit Dateigröße und Erstellungsdatum angezeigt. Geschützte Dateien sind mit einem „S" gekennzeichnet.
Danach wird der Platzbedarf für ungeschützte und geschützte Dateien sowie der freie Diskettenbereich angezeigt. Dieses Inhaltsverzeichnis ist wesentlich umfangreicher, als das MSX DOS Directory.

<F 4>-Monit. Schaltet in den MONITOR-Modus um.

<F 5>-Disk Schaltet in den DISK-Modus um.

<F 6>-Prot Diese Option schützt eine Datei auf der Diskette. Es wird nach dem Dateinamen gefragt, welcher bei der Eingabe bereits zu Großbuchstaben generiert und formatiert wird. <BS> löscht die gerade aktive Eingabe, und <ESC> bricht diese ab.

<F 7>-Unprot Hebt einen Dateischutz wieder auf. Es wird hier verfahren wie bei der Option <F 6>.

<F 8>-Datum Das anfangs eingegebene Datum kann geändert werden. Nach der Auswahl ist einfach das neue Datum einzugeben.

<F 9>-Laufw. Das aktuelle Laufwerk (A oder B) kann geändert werden.

<F10>-Ende Bringt die Funktionstastenanzeige wieder in die Normalstellung und beendet das Programm.

Bei den Auswahlen <Name>, <Prot> und <Unprot> wird automatisch das anfangs eingegebene oder geänderte Datum mit auf der Diskette eingetragen.

MONITOR-Modus

Nach dieser Auswahl (<F4> im Hauptmenü) schaltet der Rechner in den Monitor-Modus, welcher den direkten Zugriff auf die einzelnen Sektoren der Diskette erlaubt. Hier stehen wieder 5 Auswahlen zur Verfügung:

<F 1>-LOAD Diese Auswahl muß zuerst ausgeführt werden, um einen Sektor von der Diskette zu laden. Es wird angezeigt, welcher der kleinste und größte ladbare Sektor auf der Diskette ist. (Dies ist abhängig von der Formatierungsart der Diskette, welche jedoch von Disketten-Monitor erkannt wird).
Die niedrigen Sektoren sind für das System belegt und beinhalten Directory, FAT-Liste und Hauptladesatz. Diese Sektoren können auch geladen werden.

Dies ist jedoch gefährlich, da eine Veränderung in diesen Sektoren ohne genaue Systemkenntisse den ganzen Diskettenzugriff beinflussen und somit die Disketten unbrauchbar machen könnten. Es sei deshalb jedem selbst überlassen, durch Programmanalyse herauszufinden, wie sich ein Zugriff auf diese Sektoren realisieren läßt, um zu vermeiden, daß unbeabsichtigt eine Diskette zerstört wird.

<F 2>-SAVE Sichert einen zuvor geladenen Sektor wieder auf die Diskette zurück.

<F 3>-EDIT Erlaubt es, in einem zuvor geladenen Sektor Veränderungen vorzunehmen. Nachdem der Editor sich aktiv gemeldet hat, ist die Byte-Nummer im Sektor sowie der neue Inhalt durch Komma getrennt einzugeben (z. B. 100,65 oder 240,&H20), und der Rechner wird diesen Eintrag bestätigen. Um den Editiervorgang zu beenden, wird nur ein Komma und anschließend <RETURN> eingegeben. ACHTUNG: Ein editierter Sektor wird erst durch SAVE (Option <F 2>) auf die Diskette übertragen.

<F 4>-LIST Gibt den Inhalt oder einen Teil des Inhaltes eines Sektors wahlweise auf dem Monitor oder Drucker aus. Der Listvorgang kann jederzeit durch die <ESC>-Taste abgebrochen werden.

<F 5>-Ende Beendet den MONITOR-Modus und springt ins Hauptmenü zurück.

DISK-MODUS

Dieser Modus erlaubt es Ihnen, die direkten Diskettenbefehle aus dem MSX DISK BASIC über Hilfseingebemasken auszuführen. Hierbei steht Ihnen folgendes Menü zur Verfügung:

<F 1>-LFILES gibt Diskettenstatus und Diskteninhalt wie aus dem Hauptmenü über den Drucker aus. Hierbei wird das aktuelle Laufwerk (Option <F 9> des Hauptmenüs) benutzt.

<F 2>-FORMAT formatiert eine Diskette.

<F 3>-NAME benennt eine Datei auf der Diskette im aktuellen Laufwerk um.

<F 4>-KILL löscht eine Datei auf der Diskette im aktuellen Laufwerk.

<F 5>-COPY copiert eine Datei von einer Diskette auf die andere. Bei Verwendung eines Laufwerkes ist bei der Frage nach der Originaldatei die Laufwerksbezeichnung A: und bei der Copydatei die Laufwerksbezeichnung B: vorzugeben. Sie erhalten jetzt Anweisungen zum Diskettenwechsel.

Nach der Ausführung eines jeden Befehls im DISK-Modus erfolgt der Rücksprung ins Hauptmenü.

Bei den Anweisungen <F3>-NAME, <F4>-KILL und <F5>-COPY sind die Wild Cards „*" und „?" erlaubt. Bei allen anderen Anweisungen sind diese Anweisungen nicht gestattet.

DISKETTEN-MONITOR wurde mit dem CE-TEC-DPF 550 Diskettenlaufwerk entwickelt.

```
100 '*********************************
110 '*      DISKETTEN  MONITOR       *
120 '*-------------------------------*
130 '*        MSX Disk Basic         *
140 '*-------------------------------*
150 '*     (c) 1985 Tronic Soft      *
160 '*                               *
170 '*  Erste  Programmzeile:  100   *
180 '*  Letzte Programmzeile: 2730   *
190 '*********************************
200 '
210 '
```

```
220 WIDTH37
230 CLEAR 200,56000!:DEFINT I-J,L:DEFSTR A-C:DEFSNG K:SCREEN0:WIDTH38
240 ON ERROR GOTO 2710
250 KEY OFF:GOSUB 2460
260 RESTORE 300:FOR I=1 TO 10:READ B:KEY I,B:NEXT
270 KEY ON:ON KEY GOSUB 730,330,400,410,1510,1570,1620,1670,1680,480
280 FOR IK=1 TO 10:KEY(IK) ON:NEXT
290 GOTO 270
300 DATA Status," Name",Inhalt,Monit.,Disk," Prot",Unprot,Datum,Laufw.,Ende
310 DATA LFILES,FORMAT,NAME,KILL,COPY
320 DATA LOAD,SAVE,EDIT,LIST,Ende
330 GOSUB 490:CLS:GOSUB 500
340 PRINT:PRINT"Neuer Diskettenname (max. 11 Zeichen)"
350 LINE INPUT C:IF C="" OR C=B THEN RETURN 270
360 IF LEN(C)>11 THEN C=LEFT$(C,11)
370 IF LEN(C)<11 THEN C=C+STRING$(11-LEN(C),32)
380 C=C+CHR$(8):FOR K1=1 TO 12:POKE K-12+K1,ASC(MID$(C,K1,1)):NEXT:POKE K+13,DL:
POKE K+14,DH
390 DSKO$ LW,I:RETURN 270
400 GOSUB 490:CLS:GOSUB 500:PRINT:PRINT:PRINT" Dateiname     erstellt    Größe":
PRINT:GOTO 1050
410 GOSUB 490:RESTORE 320:FOR I=1 TO 5:READ B:KEY I,B:NEXT:FOR I=6 TO 10:KEY I,"
 ":NEXT:KEY ON
420 CLS:PRINT"MONITOR Modus -  Sie haben direkten                    Zugriff auf
 die                        Sektoren der Diskette"
430 ON KEY GOSUB 2090,2150,2210,2300,2450
440 KEY ON
450 FOR I=1 TO 5:KEY(I) ON:NEXT
460 QK=0
470 A=INKEY$:IF QK THEN GOSUB 490:RETURN 260 ELSE 470
480 GOSUB 490:DEFUSR0=&H3E:KEY ON:X=USR0(1):KEY ON:END
490 KEY OFF:FOR IK=1 TO 10:KEY(IK)OFF:NEXT:RETURN
500 GOSUB 610:FOR I=NF*SF+1 TO NF*SF+DS
510 POKE &HF351,193:POKE &HF352,218:A=DSKI$(LW,I)
520 FOR K=56012! TO 56492! STEP 32:IF PEEK(K-11)=0 THEN 550
530 IF PEEK(K)=8 THEN B="":FOR J=0 TO 10:B=B+CHR$(PEEK(K-11+J)):NEXT:GOTO 560
540 NEXT K,I
550 IF PV THEN LPRINT"Diskette hat keinen Namen" ELSE PRINT "Diskette hat keinen
 Namen":RETURN
560 IF PV THEN LPRINT"Diskettenname: ";B;" "; ELSE PRINT"Diskettenname: ";B;" ";

570 A2=BIN$(PEEK(K+13)):A1=BIN$(PEEK(K+14)):IF LEN(A1)<8 THEN A1=STRING$(8-LEN(A
1),"0")+A1
580 IF LEN(A2)<8 THEN A2=STRING$(8-LEN(A2),"0")+A2
590 A=A1+A2:IT=VAL("&B"+RIGHT$(A,5)):IM=VAL("&B"+MID$(A,8,4)):IJ=1980+VAL("&B"+L
EFT$(A,7))
600 IF PV THEN LPRINTUSING"##.";IT,IM;:LPRINTUSING"####";IJ:RETURN ELSE PRINTUSI
NG"##.";IT,IM;:PRINTUSING"####";IJ:RETURN
610 POKE &HF351,193:POKE &HF352,218:A=DSKI$(LW,1):IX=PEEK(56001!)AND 15
620 ON IX-7 GOTO 630,640,650,660,670,680,690,700
630 RD=112:SF=2:ST=9:NS=1:TS=80:NF=2:SC=2:GOTO 710
640 RD=112:SF=3:ST=9:NS=2:TS=80:NF=2:SC=2:GOTO 710
650 RD=112:SF=1:ST=8:NS=1:TS=80:NF=2:SC=2:GOTO 710
660 RD=112:SF=2:ST=8:NS=2:TS=80:NF=2:SC=2:GOTO 710
670 RD=64:SF=2:ST=9:NS=1:TS=40:NF=2:SC=1:GOTO 710
680 RD=112:SF=2:ST=9:NS=2:TS=40:NF=2:SC=2:GOTO 710
690 RD=64:SF=1:ST=8:NS=1:TS=40:NF=2:SC=1:GOTO 710
700 RD=112:SF=1:ST=8:NS=2:TS=40:NF=2:SC=2:GOTO 710
710 SB=512:IF RD=112 THEN DS=7 ELSE DS=4
720 RETURN
730 GOSUB 490:POKE &HF351,193:POKE &HF352,218:A=DSKI$(LW,0):B="":FOR K=56004! TO
 56011!:B=B+CHR$(PEEK(K)):NEXT:IF PV THEN RETURN
740 CLS:PRINT"Diskette formatiert durch ";B
750 GOSUB 500
760 PRINT:PRINTUSING"###";NS;:PRINT" Diskettenseite(n)"
770 PRINTUSING"###";TS;:PRINT" Spuren pro Seite"
780 PRINTUSING"###";ST;:PRINT" Sektoren pro Spur"
790 PRINTUSING"###";SB;:PRINT" Bytes pro Sektor"
800 PRINTUSING"###";RD;:PRINT" mögliche Einträge im                    Hauptverzei
chnis.":PRINT
810 PRINTUSING"Gesamtkapazität        : ###### Bytes";NS*TS*ST*SB
820 PRINTUSING"Durch System belegt    : ###### Bytes";SF*NF*SB+SB+SB*DS
830 PRINTUSING"Datenbereich           : ###### Bytes";NS*TS*ST*SB-(SF*NF*SB+SB+S
B*DS)
840 GOSUB 860
850 RETURN
860 GOSUB 610:FD=0:FB=0:GD=0:GB=0:FOR I=NF*SF+1 TO NF*SF+DS
870 POKE &HF351,193:POKE &HF352,218:A=DSKI$(LW,I)
880 FOR K=56012! TO 56492! STEP 32:IF PEEK(K-11)=0 THEN 940
890 IF PEEK(K-11)=229 THEN 930
900 IF PEEK(K)=8 THEN 930
910 IF PEEK(K)=0 OR PEEK(K)=32 OR PEEK(K)=1 THEN 1010
920 IF PEEK(K)AND2=2 OR PEEK(K)AND 4=4 THEN 1030
930 NEXT K,I
940 IF PV=0 THEN PRINT:PRINTUSING"### Benutzerdatei(en)     ###### Bytes";FD,FB
950 IF PV THEN LPRINT:LPRINTUSING"### Benutzerdatei(en)     ###### Bytes";FD,FB
960 IF PV=0 THEN PRINTUSING"### geschützte Datei(en) ###### Bytes";GD,GB
970 IF PV THEN LPRINTUSING"### geschuetzte Datei(en) ###### Bytes";GD,GB
980 IF PV=0 THEN PRINT:PRINTUSING"freier Diskettenbereich: ###### Bytes";DSKF(LW
)*512
990 IF PV THEN LPRINT:LPRINTUSING"freier Diskettenbereich:  ###### Bytes";DSKF(L
W)*512
1000 IF PV THEN LPRINT:LPRINT:LPRINT:LPRINT:GOTO 1850 ELSE RETURN 270
1010 B="":FOR K1=K+20 TO K+17 STEP -1:B1=BIN$(PEEK(K1)):IF LEN(B1)<8 THEN B1=STR
ING$(8-LEN(B1),"0")+B1
1020 B=B+B1:NEXT K1:FD=FD+1:FB=FB+VAL("&B"+B):GOTO 930
1030 B="":FOR K1=K+20 TO K+17 STEP -1:B1=BIN$(PEEK(K1)):IF LEN(B1)<8 THEN B1=STR
ING$(8-LEN(B1),"0")+B1
1040 B=B+B1:NEXT K1:GD=GD+1:GB=GB+VAL("&B"+B):GOTO 930
1050 FD=0:FB=0:GD=0:GB=0:FOR I=NF*SF+1 TO NF*SF+DS
1060 POKE &HF351,193:POKE &HF352,218:A=DSKI$(LW,I)
1070 FOR K=56012! TO 56492! STEP 32:IF PEEK(K-11)=0 THEN 940
1080 IF PEEK(K-11)=229 THEN 1120
1090 IF PEEK(K)=8 THEN 1120
1100 IF PEEK(K)=0 OR PEEK(K)=32 OR PEEK(K)=1 THEN 1130
1110 IF PEEK(K)AND2=2 OR PEEK(K)AND 4=4 THEN 1150
1120 NEXT K,I
1130 B="":FOR K1=K+20 TO K+17 STEP -1:B1=BIN$(PEEK(K1)):IF LEN(B1)<8 THEN B1=STR
ING$(8-LEN(B1),"0")+B1
1140 B=B+B1:NEXT K1:FD=FD+1:FB=FB+VAL("&B"+B):GOSUB 1170:IF PV THEN LPRINT:GOTO
1120 ELSE PRINT:GOTO 1120
1150 B="":FOR K1=K+20 TO K+17 STEP -1:B1=BIN$(PEEK(K1)):IF LEN(B1)<8 THEN B1=STR
ING$(8-LEN(B1),"0")+B1
1160 B=B+B1:NEXT K1:GD=GD+1:GB=GB+VAL("&B"+B):GOSUB 1170:IF PV THEN LPRINT"S":GO
TO 1120 ELSE PRINT"S":GOTO 1120
1170 B2="":FOR K1=K-11 TO K-1:B2=B2+CHR$(PEEK(K1)):NEXT
1180 A2=BIN$(PEEK(K+13)):A1=BIN$(PEEK(K+14)):IF LEN(A1)<8 THEN A1=STRING$(8-LEN(
A1),"0")+A1
1190 IF LEN(A2)<8 THEN A2=STRING$(8-LEN(A2),"0")+A2
1200 A=A1+A2:IT=VAL("&B"+RIGHT$(A,5)):IM=VAL("&B"+MID$(A,8,4)):IJ=1980+VAL("&B"+
LEFT$(A,7))
1210 IF PV=0 THEN PRINTLEFT$(B2,8)+"."+RIGHT$(B2,3)+"  ";:PRINTUSING"##.";IT,IM;
```

```
:PRINTUSING"####";IJ;:PRINTUSING"  ######  ";VAL("&B"+B);:RETURN
1220 IF PV THEN LPRINTLEFT$(B2,8)+"."+RIGHT$(B2,3)+"  ";:LPRINTUSING"##.";IT,IM;
:LPRINTUSING"####";IJ;:LPRINTUSING"  ######  ";VAL("&B"+B);:RETURN
1230 LOCATE 0,2:PRINTSTRING$(20,32):C7=CHR$(LW+64)+":":LOCATE 0,2:PRINTC7
1240 A=INKEY$:IF A=""THEN 1240
1250 IF A=CHR$(27) THEN QK=-1:RETURN
1260 IF A=CHR$(8) THEN 1230
1270 IF A=CHR$(13) THEN 1410
1280 IF A=CHR$(46) THEN 1320
1290 C7=C7+CHR$(ASC(A)+32*(ASC(A)>95)):LOCATE 0,2:PRINTC7
1300 IF LEN(C7)=10 THEN 1330
1310 GOTO 1240
1320 IF LEN(C7)<10 THEN C7=C7+STRING$(10-LEN(C7),32)
1330 C7=C7+".":LOCATE 0,2:PRINTC7
1340 A=INKEY$:IF A=""THEN 1340
1350 IF A=CHR$(27) THEN QK=-1:RETURN
1360 IF A=CHR$(8) THEN 1230
1370 IF A=CHR$(13) THEN 1410
1380 C7=C7+CHR$(ASC(A)+32*(ASC(A)>95)):LOCATE 0,2:PRINTC7
1390 IF LEN(C7)=14 THEN 1410
1400 GOTO 1340
1410 IF LEN(C7)<14 THEN C7=C7+STRING$(14-LEN(C7),32)
1420 C7=MID$(C7,3,8)+MID$(C7,12,3):RETURN
1430 GOSUB 610:FOR I=NF*SF+1 TO NF*SF+DS
1440 POKE &HF351,193:POKE &HF352,218:A=DSKI$(LW,I)
1450 FOR K=56012! TO 56492! STEP 32:IF PEEK(K-11)=0 THEN 1500
1460 IF PEEK(K-11)=229 THEN 1490
1470 B2="":FOR K1=K-11 TO K-1:B2=B2+CHR$(PEEK(K1)):NEXT
1480 IF B2=C7 THEN QK=0:RETURN
1490 NEXT K,I
1500 BEEP:PRINT:BEEP:PRINT"Datei existiert nicht auf der Diskette":BEEP:QK=-1:RE
TURN
1510 GOSUB 490:RESTORE 310:FOR I=1 TO 5:READ B:KEY I,B:NEXT:FOR I=6 TO 10:KEY I,
" ":NEXT:KEY ON
1520 ON KEY GOSUB 1690,1860,1900,1970,2030
1530 KEY ON
1540 FOR I=1 TO 5:KEY(I) ON:NEXT
1550 QK=0
1560 IF QK THEN GOSUB 490:RETURN 260 ELSE 1560
1570 GOSUB 490:CLS:PRINT"Protektion für Datei vergeben":QK=0:GOSUB 1230:IF QK TH
EN RETURN 270
1580 GOSUB 1430:IF QK THEN RETURN 270
1590 IF PEEK(K)AND2=2 OR PEEK(K)AND4=4 THEN BEEP:PRINT:BEEP:PRINT"Datei ist bere
its geschützt":RETURN 270
1600 POKE K+13,DL:POKE K+14,DH
1610 POKE K,2:DSKO$LW,I:PRINT:PRINT"Datei ist jetzt geschützt":RETURN 270
1620 GOSUB 490:CLS:PRINT"Protektion für Datei aufheben":QK=0:GOSUB 1230:IF QK TH
EN RETURN 270
1630 GOSUB 1430:IF QK THEN RETURN 270
1640 IF PEEK(K)=0 OR PEEK(K)=32 OR PEEK(K)=1 THEN BEEP:PRINT:BEEP:PRINT"Datei is
t bereits ungeschützt":RETURN 270
1650 POKE K+13,DL:POKE K+14,DH
1660 POKE K,0:DSKO$LW,I:PRINT:PRINT"Datei ist jetzt ungeschützt":RETURN 270
1670 GOSUB 490:GOSUB 2500:RETURN270
1680 GOSUB 490:GOSUB 2680:RETURN270
1690 GOSUB 490:PRINT:PRINT"Ist der Drucker eingeschaltet? (J/N)"
1700 A=INKEY$:IF A=""THEN1700 ELSE A=CHR$(ASC(A)+32*(ASC(A)>96)):IF INSTR("JN",A
)=0 THEN 1700
1710 ON INSTR("JN",A) GOTO 1730,1720
1720 QK=-1:RETURN
1730 LPRINT"Inhaltsverzeichnis vom ";:LPRINT USING"##.";DT,DM;:LPRINT USING"####
```



```
";DJ:LPRINT
1740 PV=-1:GOSUB 500
1750 GOSUB 730:LPRINT"Diskette formatiert durch ";B
1760 LPRINT:LPRINTUSING"###";NS;:LPRINT" Diskettenseite(n)"
1770 LPRINTUSING"###";TS;:LPRINT" Spuren pro Seite"
1780 LPRINTUSING"###";ST;:LPRINT" Sektoren pro Spur"
1790 LPRINTUSING"###";SB;:LPRINT" Bytes pro Sektor"
1800 LPRINTUSING"###";RD;:LPRINT" moegliche Eintraege im":LPRINT"    Hauptverzei
chnis.":LPRINT
1810 LPRINTUSING"Gesamtkapazitaet       : ###### Bytes";NS*TS*ST*SB
1820 LPRINTUSING"Durch System belegt    : ###### Bytes";SF*NF*SB+SB+SB*DS
1830 LPRINTUSING"Datenbereich           : ###### Bytes";NS*TS*ST*SB-(SF*NF*SB+SB
+SB*DS):LPRINT
1840 LPRINT:LPRINT" Dateiname     erstellt   Groesse":LPRINT:GOTO 1050
1850 PV=0:QK=-1:RETURN
1860 GOSUB 490:PRINT:PRINT"Haben Sie sich vergewissert, daß die  Diskette keine
Dateien enthält die    noch benötigt werden? (J/N)"
1870 A=INKEY$:IF A=""THEN1870 ELSE A=CHR$(ASC(A)+32*(ASC(A)>96)):IF INSTR("JN",A
)=0 THEN 1870
1880 ON INSTR("JN",A) GOTO 1890,1720
1890 CALL FORMAT:QK=-1:RETURN
1900 GOSUB 490:PRINT:PRINT"Bisheriger Dateiname ";CHR$(LW+64)+":";:LINE INPUT A1
$
1910 PRINT:PRINT"Neuer      Dateiname ";CHR$(LW+64)+":";:LINE INPUT A2$
1920 A1$=CHR$(LW+64)+":"+A1$:A2$=CHR$(LW+64)+":"+A2$
1930 PRINT:PRINT"Sind Sie sicher, daß Sie die Datei wievorstehend umbenennen wol
len? (J/N)"
1940 A=INKEY$:IF A=""THEN1940 ELSE A=CHR$(ASC(A)+32*(ASC(A)>96)):IF INSTR("JN",A
)=0 THEN 1940
1950 ON INSTR("JN",A) GOTO 1960,1720
1960 NAME A1$ AS A2$:QK=-1:RETURN
1970 GOSUB 490:PRINT:PRINT"Zu löschende Datei ";CHR$(LW+64)+":";:LINE INPUT A1$
1980 A1$=CHR$(LW+64)+":"+A1$
1990 PRINT:PRINT"Sind Sie sicher, daß Sie die vor-     stehende Datei löschen wo
llen? (J/N)"
2000 A=INKEY$:IF A=""THEN2000 ELSE A=CHR$(ASC(A)+32*(ASC(A)>96)):IF INSTR("JN",A
)=0 THEN 2000
2010 ON INSTR("JN",A) GOTO 2020,1720
2020 KILL A1$:QK=-1:RETURN
2030 GOSUB 490:PRINT:PRINT"Name der Originaldatei: ";:LINE INPUT A1$
2040 PRINT:PRINT"Name der     Copydatei: ";:LINE INPUT A2$
2050 PRINT:PRINT"Sind Sie sicher, daß Sie die Datei wievorstehend copiert werden
 soll? (J/N)"
2060 A=INKEY$:IF A=""THEN2060 ELSE A=CHR$(ASC(A)+32*(ASC(A)>96)):IF INSTR("JN",A
)=0 THEN 2060
2070 ON INSTR("JN",A) GOTO 2080,1720
2080 COPY A1$ TO A2$:QK=-1:RETURN
2090 GOSUB 490:GOSUB 610:CLS:PRINT"Welcher Sektor soll geladen werden?":LS=SF*NF
+DS+1:GS=NS*TS*ST-1:PRINTUSING"Kleinster Sektor ###";LS
2100 PRINTUSING"Größter   Sektor ###";GS:PRINT
2110 LINE INPUT "Bitte Sektor wählen : ";AS:SN=VAL(AS):IF SN>GS THEN RETURN 430
2120 IF RIGHT$(AS,1)=CHR$(126) THEN 2130 ELSE IF SN<LS THEN RETURN 430
2130 IF RIGHT$(AS,1)=CHR$(27) THEN RETURN 430
2140 POKE &HF351,193:POKE &HF352,218:A=DSKI$(LW,SN):CLS:PRINTUSING"Sektor ### is
t geladen";SN:SL=-1:RETURN 430
2150 IF NOT SL THEN BEEP:PRINT"Es ist kein Sektor geladen":RETURN 430
2160 GOSUB 490:CLS:PRINTUSING"Sektor ### ist geladen";SN:PRINT:PRINT"Soll dieser
 Sektor gesichert werden?  (J/N)"
2170 A=INKEY$:IF A=""THEN2170 ELSE A=CHR$(ASC(A)+32*(ASC(A)>96)):IF INSTR("JN",A
)=0 THEN 2170
```

```
2180 ON INSTR("JN",A) GOTO 2200,2190
2190 RETURN 430
2200 DSKO$ LW,SN:RETURN 430
2210 IF NOT SL THEN BEEP:PRINT"Es ist kein Sektor geladen":RETURN 430
2220 GOSUB 490:PRINT:PRINTUSING"EDITOR für Sektor ### ist aktiv";SN
2230 PRINTCHR$(LW+64)+": Sektor";SN;:INPUT ">";BT%,IH
2240 IF BT%<1 OR BT%>512 THEN PRINT:PRINT"EDITOR beendet":PRINT:RETURN 430
2250 POKE 56000!+BT%,IH
2260 IF PEEK(56000!+BT%)<32 THEN AW=CHR$(32) ELSE AW=CHR$(PEEK(56000!+BT%))
2270 AH=HEX$(PEEK(56000!+BT%)):IF LEN(AH)<2 THEN AH="&h0"+AH ELSE AH="&h"+AH
2280 PRINT USING"###/###   ###   &    &";SN,BT%,PEEK(56000!+BT%),AH,AW
2290 GOTO 2230
2300 IF NOT SL THEN BEEP:PRINT"Es ist kein Sektor geladen":RETURN 430
2310 GOSUB 490:CLS:PRINT"Ausgabe am Monitor oder Drucker? (M/D)"
2320 A=INKEY$:IF A=""THEN2320 ELSE IF A=CHR$(27) THEN RETURN 430 ELSE A=CHR$(ASC
(A)+32*(ASC(A)>96)):IF INSTR("MD",A)=0 THEN 2320
2330 ON INSTR("MD",A) GOTO 2350,2340
2340 PV=-1:GOTO 2360
2350 PV=0
2360 LINE INPUT "Ab welcher Stelle ?    (1-511) ";AS:SS=VAL(AS)
2370 LINE INPUT "Bis zu welcher Stelle? (2-512) ";AE:ES=VAL(AE)
2380 IF ES<SS OR SS<1 OR ES>512 THEN RETURN 430
2390 IF PV THEN LPRINT USING"Inhalt des Sektors ###";SN:LPRINT ELSE PRINT:PRINT:
PRINT USING"Inhalt des Sektors ###";SN:PRINT
2400 FOR LF=SS TO ES:A=INKEY$:IF A=CHR$(27) THEN RETURN 430
2410 IF PEEK(56000!+LF)<32 THEN AW=CHR$(32) ELSE AW=CHR$(PEEK(56000!+LF))
2420 AH=HEX$(PEEK(56000!+LF)):IF LEN(AH)<2 THEN AH="&h0"+AH ELSE AH="&h"+AH
2430 IF PV THEN LPRINT USING"###/###   ###   &    &";SN,LF,PEEK(56000!+LF),AH,AW
 ELSE PRINT USING"###/###   ###   &    &";SN,LF,PEEK(56000!+LF),AH,AW
2440 NEXT:RETURN 430
2450 GOSUB 490:PV=0:SL=0:RETURN 260
2460 CLS:PRINT"          DISKETTEN MONITOR"
2470 PRINT"          -----------------"
2480 PRINT:PRINT"      (c) 1985 by Tronic Soft"
2490 GOSUB 2500:GOSUB 2680:RETURN
2500 PRINT:PRINT:PRINT"Bitte Datum eingeben (TT.MM.JJJJ)"
2510 LINE INPUT B$
2520 IF INSTR(B,".")=0 THEN 2460
2530 P=INSTR(B,"."):DT=VAL(LEFT$(B,P-1))
2540 IF INSTR(P+1,B,".")=0 THEN 2460
2550 P2=INSTR(P+1,B,"."):DM=VAL(MID$(B,P+1,P2-P-1))
2560 DJ=VAL(RIGHT$(B,4))
2570 IF DT<1 OR DT>31 THEN 2460
2580 IF DM<1 OR DM>12 THEN 2460
2590 IF DJ<1980 OR DM>2099 THEN 2460
2600 B="":B1=BIN$(DJ-1980):IF LEN(B1)<7 THEN B1=STRING$(7-LEN(B1),"0")+B1
2610 B=B+B1
2620 B1=BIN$(DM):IF LEN(B1)<4 THEN B1=STRING$(4-LEN(B1),"0")+B1
2630 B=B+B1
2640 B1=BIN$(DT):IF LEN(B1)<5 THEN B1=STRING$(5-LEN(B1),"0")+B1
2650 B=B+B1
2660 DH=VAL("&B"+LEFT$(B,8)):DL=VAL("&B"+RIGHT$(B,8))
2670 RETURN
2680 PRINT:PRINT"Welches Laufwerk? (A/B)"
2690 A=INKEY$:IF A=""THEN2690 ELSE A=CHR$(ASC(A)+32*(ASC(A)>96)):IF INSTR("AB",A
)=0 THEN 2690
2700 LW=INSTR("AB",A):RETURN
2710 BEEP:PRINT:BEEP:PRINT"Diskettenfehler, Befehl konnte nicht  ordnungsgemäß a
usgeführt werden.":BEEP
2720 ON ERROR GOTO 2710
2730 RESUME 260
```


Dieses Programm erweitert den Befehlssatz des C-64 um 38 Musik- und Grafikbefehle

MULTI BASIC V3

Eine Basic-Erweiterung für den Commodore 64

Sprites, Musik und Graphik auf dem 64'er ohne POKE? Diskettenbehandlung ohne OPEN... und LOAD "$",8? Unmöglich! Nein, man braucht nur das richtige Werkzeug. MULTI BASIC V3 ist eines dieser Werkzeuge (engl. tools).

MULTI BASIC V3 erweitert das Standard BASIC des C64 um 38 neue Befehle für Sprites, Musik, hochauflösende Graphik und Diskette. Alle diese Befehle und Funktionen können wie die Originalbefehle abgekürzt werden; alles läuft wie gewohnt ab.

Das ca. 4k lange Maschinenprogramm liegt von 49152 bis 52674 im Speicher (C000-CDC2). Nach der Initialisierung wurde der Anfang des Basicspeichers auf 16384 gesetzt. Es stehen also nur noch 24k Speicher für Programme zur Verfügung, dies dürfte aber ausreichen.

Der Graphikspeicher beginnt ab 8192 bis 16383. Bei der hochauflösenden Graphik beginnt der Video-Speicher bei 2048.

```
0 REM"*************************************
1 REM"*        MULTI BASIC V3             *
2 REM"*                                   *
3 REM"* (C)1985 / WRITTEN BY P.WITZEL     *
4 REM"*************************************
5 :
6 PRINT"■":PRINT,"  MULTI BASIC V3"
7 PRINT "■          1985 BY PETER WITZEL"
10 DIM H(75):FORI=0TO9
20 H(48+I)=I:H(65+I)=I+10:NEXT
30 FORI=49152 TO 52674:READ A$
40 H=ASC(LEFT$(A$,1)):L=ASC(RIGHT$(A$,1))
50 D=H(H)*16+H(L):S=S+D:POKEI,D
60 A=A+1:IFA<20 THEN NEXT:A=-1
65 PRINT "■■■■■■■■ZEILE :"1000+Z
70 READ V:Z=Z+1:IF V=S THEN 85
80 PRINT "FEHLER IN ZEILE "999+Z"!":END
85 IF A<0 THEN 91
90 S=0:A=0:NEXT
91 SYS 49673
92 END
93 :
94 :
95 :
1000 DATA AD,18,D0,09,08,8D,18,D0,AD,11,D0,09,20,8D,11,D0,60,A9,20,85,  2030
1001 DATA 25,A9,00,85,24,A2,20,A0,00,91,24,88,D0,FB,E6,25,CA,D0,F4,60,  2522
1002 DATA A2,00,9D,00,08,9D,00,09,9D,00,0A,9D,00,0B,E8,D0,F1,4C,A3,CD,  1953
1003 DATA 20,4E,C0,11,22,91,22,60,20,4E,C0,49,FF,31,22,91,22,60,20,79,  1769
1004 DATA 00,20,EB,B7,86,23,A5,14,29,07,AA,45,14,85,14,A9,00,85,22,38,  1656
1005 DATA 6A,CA,10,FC,AA,A5,23,29,07,A8,45,23,85,23,4A,4A,65,23,4A,4A,  1866
1006 DATA 66,22,4A,66,22,85,23,A5,14,65,22,85,22,A5,15,65,23,09,20,85,  1497
1007 DATA 23,8A,60,AD,11,D0,29,DF,8D,11,D0,AD,18,D0,29,F7,8D,18,D0,A9,  2532
1008 DATA 93,20,D2,FF,A9,00,85,C6,60,20,EB,B7,A5,14,8D,00,CE,A5,15,8D,  2549
1009 DATA 01,CE,8E,02,CE,20,FD,AE,20,EB,B7,A5,14,8D,03,CE,A5,15,8D,04,  2332
1010 DATA CE,8E,05,CE,4C,74,CD,A9,FF,8D,0D,CE,8D,0C,CE,A9,00,8D,08,CE,  2623
1011 DATA 8D,0B,CE,8D,07,CE,38,AD,05,CE,ED,02,CE,8D,06,CE,B0,07,CE,07,  2346
1012 DATA CE,CE,08,CE,38,AD,03,CE,ED,00,CE,8D,09,CE,AD,04,CE,ED,01,CE,  2690
1013 DATA 8D,0A,CE,B0,1D,CE,0B,CE,30,18,0E,09,CE,2E,0A,CE,2E,0B,CE,0E,  1825
1014 DATA 06,CE,2E,07,CE,2E,08,CE,4E,0D,CE,6E,0C,CE,AD,08,CE,4A,6A,4D,  2000
1015 DATA 07,CE,30,0A,AD,0B,CE,4A,6A,4D,0A,CE,10,D4,AD,00,CE,8D,12,CE,  2106
1016 DATA AD,01,CE,8D,13,CE,AD,02,CE,8D,22,CE,A9,80,8D,11,CE,8D,21,CE,  2549
1017 DATA 0A,8D,10,CE,8D,20,CE,8D,23,CE,20,C9,C1,AC,22,CE,AD,20,CE,18,  2407
1018 DATA 6D,06,CE,8D,20,CE,AD,21,CE,6D,07,CE,8D,21,CE,AD,22,CE,6D,08,  2344
```

```
1019 DATA CE,8D,22,CE,AE,12,CE,18,AD,10,CE,6D,09,CE,8D,10,CE,AD,11,CE, 2487
1020 DATA 6D,0A,CE,8D,11,CE,AD,12,CE,6D,0B,CE,8D,12,CE,AD,13,CE,6D,0B, 2295
1021 DATA CE,8D,13,CE,CC,22,CE,D0,05,EC,12,CE,F0,03,20,C9,C1,AD,0C,CE, 2749
1022 DATA D0,09,AD,0D,CE,D0,01,60,CE,0D,CE,CE,0C,CE,4C,61,C1,AD,12,CE, 2526
1023 DATA 85,14,AD,13,CE,85,15,AE,22,CE,20,54,C0,11,22,91,22,60,AD,12, 1944
1024 DATA CE,85,14,AD,13,CE,85,15,AE,22,CE,20,54,C0,49,FF,31,22,91,22, 2223
1025 DATA 60,A9,DE,8D,5F,C1,8D,B3,C1,20,A9,C0,A9,C9,8D,5F,C1,8D,B3,C1, 3134
1026 DATA 60,A9,BB,8D,04,03,A9,C6,8D,05,03,A9,31,8D,06,03,A9,C8,8D,07, 2001
1027 DATA 03,20,66,C8,20,F1,CA,A9,00,85,B6,A9,68,8D,18,03,A9,CD,8D,19, 2277
1028 DATA 03,68,68,A2,80,4C,8B,E3,20,00,C0,18,20,2C,CD,4C,63,C2,EA,20, 2107
1029 DATA 11,C0,4C,63,C2,20,79,00,20,9E,B7,8A,20,28,C0,4C,64,C2,20,F5, 2153
1030 DATA C1,4C,64,C2,20,8F,C0,38,20,2C,CD,EA,60,20,79,00,20,9E,B7,8E, 2265
1031 DATA 20,D0,20,FD,AE,20,9E,B7,8E,21,D0,20,FD,AE,20,9E,B7,8E,86,02, 2565
1032 DATA 4C,64,C2,EA,20,79,00,20,9E,B7,86,02,20,FD,AE,20,9E,B7,A4,02, 2264
1033 DATA 18,20,F0,FF,20,FD,AE,20,B5,AA,4C,64,C2,EA,A9,00,85,C6,A5,C6, 2860
1034 DATA F0,FC,4C,63,C2,EA,A2,19,A9,00,CA,9D,00,D4,D0,FA,4C,63,C2,EA, 3083
1035 DATA 20,79,00,20,9E,B7,8E,18,D4,A9,FF,8D,02,D4,8D,09,D4,8D,10,D4, 2414
1036 DATA 4C,64,C2,20,79,00,20,9E,B7,E0,04,90,03,4C,48,B2,CA,F0,08,CA, 2249
1037 DATA F0,0A,A0,0E,84,02,60,A0,00,84,02,60,A0,07,84,02,60,EA,20,D3, 1918
1038 DATA C2,20,FD,AE,20,9E,B7,8A,A4,02,99,05,D4,20,FD,AE,20,9E,B7,8A, 2670
1039 DATA A4,02,99,06,D4,4C,64,C2,EA,20,D3,C2,20,FD,AE,20,9E,B7,A4,02, 2576
1040 DATA E0,05,B0,B9,E0,01,F0,18,E0,02,F0,19,E0,03,F0,1A,E0,04,F0,1B, 2558
1041 DATA B9,00,CF,AA,CA,8A,99,04,D4,4C,64,C2,A9,11,4C,51,C3,A9,21,4C, 2457
1042 DATA 51,C3,A9,41,4C,51,C3,A9,81,99,00,CF,99,04,D4,4C,64,C2,EA,20, 2525
1043 DATA D3,C2,20,FD,AE,20,8A,AD,20,F7,B7,A4,02,A5,14,99,00,D4,A5,15, 2571
1044 DATA 99,01,D4,4C,64,C2,EA,20,79,00,20,9E,B7,E0,08,90,03,4C,DD,C2, 2366
1045 DATA 86,02,20,FD,AE,20,9E,B7,8A,A6,02,9D,F8,07,99,F8,0B,4C,64,C2, 2468
1046 DATA 20,79,00,20,9E,B7,E0,08,90,03,4C,DD,C2,8E,E9,07,A0,00,A9,01, 2108
1047 DATA 85,02,CC,E9,07,F0,06,06,02,C8,4C,AE,C3,A5,02,0D,15,D0,8D,15, 2049
1048 DATA D0,4C,64,C2,20,79,00,20,9E,B7,E0,08,90,03,4C,DD,C2,8E,E9,07, 2356
1049 DATA 8E,EA,07,20,FD,AE,20,EB,B7,8E,EB,07,0E,E9,07,20,07,C4,A5,15, 2351
1050 DATA C9,00,D0,2D,A5,02,49,FF,2D,10,D0,8D,10,D0,AC,E9,07,A5,14,99, 2333
1051 DATA 00,D0,AD,EB,07,99,01,D0,4C,64,C2,A0,00,A9,01,85,02,CC,EA,07, 2265
1052 DATA F0,06,06,02,C8,4C,0D,C4,60,A5,02,0D,10,D0,8D,10,D0,4C,F6,C3, 2121
1053 DATA EA,20,79,00,20,9E,B7,E0,08,90,03,4C,DD,C2,8E,EA,07,20,07,C4, 2248
1054 DATA A5,02,0D,1D,D0,8D,1D,D0,4C,64,C2,EA,20,79,00,20,9E,B7,E0,08, 2157
1055 DATA 90,03,4C,DD,C2,8E,EA,07,20,07,C4,A5,02,0D,17,D0,8D,17,D0,4C, 2115
1056 DATA 64,C2,EA,20,79,00,20,9E,B7,E0,08,90,03,4C,DD,C2,8E,EA,07,20, 2339
1057 DATA 07,C4,A5,02,49,FF,2D,1D,D0,8D,1D,D0,4C,64,C2,EA,20,79,00,20, 2147
1058 DATA 9E,B7,E0,08,90,03,4C,DD,C2,8E,EA,07,20,07,C4,A5,02,49,FF,2D, 2369
1059 DATA 17,D0,8D,17,D0,4C,64,C2,EA,20,79,00,20,9E,B7,E0,08,90,03,4C, 2188
1060 DATA DD,C2,8E,EA,07,20,FD,AE,20,9E,B7,AC,EA,07,8A,99,27,D0,4C,64, 2757
1061 DATA C2,20,79,00,D0,08,A9,00,8D,15,D0,4C,63,C2,20,79,00,20,9E,B7, 1997
1062 DATA E0,08,90,03,4C,DD,C2,8E,EA,07,20,07,C4,A5,02,49,FF,2D,15,D0, 2257
1063 DATA 8D,15,D0,4C,64,C2,EA,20,79,00,20,9E,AD,20,A6,B6,85,02,8E,E8, 2379
1064 DATA 07,8C,E9,07,20,FD,AE,20,9E,B7,20,BA,FF,A5,02,AE,E8,07,AC,E9, 2677
1065 DATA 07,20,BD,FF,A9,00,85,FB,A9,20,85,FC,A9,FB,A2,40,A0,3F,20,D8, 2739
1066 DATA FF,20,0F,E1,4C,64,C2,EA,20,79,00,20,9E,AD,20,A6,B6,85,02,8E, 2304
1067 DATA E8,07,8C,E9,07,20,FD,AE,20,9E,B7,A0,00,20,BA,FF,A5,02,AE,E8, 2657
1068 DATA 07,AC,E9,07,20,BD,FF,A9,00,A2,00,A0,20,20,D5,FF,20,0F,E1,4C, 2266
1069 DATA 64,C2,EA,20,79,00,20,9E,AD,20,A6,B6,85,02,8E,E8,07,8C,E9,07, 2320
1070 DATA A9,0F,85,B8,A9,08,85,BA,A9,0F,85,B9,A5,02,85,B7,AD,E8,07,85, 2527
1071 DATA BB,AD,E9,07,85,BC,20,C0,FF,A9,0F,20,C3,FF,20,0F,E1,4C,64,C2, 2708
1072 DATA 43,9D,45,C3,C1,49,44,24,4E,C4,C9,52,45,B6,57,46,ED,5C,C6,73, 2465
1073 DATA 62,47,4E,68,C7,82,6E,41,14,75,C1,0A,7C,48,6A,83,EA,20,D3,C2, 2299
1074 DATA 20,FD,AE,20,9E,B7,8E,E8,07,20,FD,AE,20,9E,AD,20,A6,B6,86,FB, 2800
1075 DATA 84,FC,A0,00,B1,FB,8D,E9,07,B9,A0,C5,CD,E9,07,F0,07,C8,C8,C8, 3187
1076 DATA C0,21,90,F1,A6,02,B9,A2,C5,8D,EB,07,B9,A1,C5,20,0E,C6,9D,00, 2649
1077 DATA D4,AD,ED,07,9D,01,D4,4C,64,C2,EA,A0,06,CC,E8,07,F0,08,4E,EB, 2773
1078 DATA 07,6A,88,4C,11,C6,48,AD,EB,07,8D,ED,07,68,60,EA,A9,08,85,BA, 2342
1079 DATA 20,B4,FF,A9,6F,85,B9,20,96,FF,20,A5,FF,20,D2,FF,C9,0D,D0,F6, 3119
```


CPU-User / C-64

```
1080 DATA 20,AB,FF,4C,63,C2,A9,40,20,90,FF,A9,24,85,FB,A9,FB,85,BB,A9, 2989
1081 DATA 00,85,BC,A9,01,85,B7,A9,08,85,BA,A9,60,85,B9,20,D5,F3,A5,BA, 2725
1082 DATA 20,B4,FF,A5,B9,20,96,FF,A9,00,85,90,A0,03,84,FB,20,A5,FF,85, 2831
1083 DATA FC,A4,90,D0,34,20,A5,FF,A4,90,D0,2D,A4,FB,88,D0,E9,A6,FC,20, 3275
1084 DATA CD,BD,A9,20,20,D2,FF,20,A5,FF,A6,90,D0,17,AA,F0,06,20,D2,FF, 2998
1085 DATA 4C,97,C6,A9,0D,20,D2,FF,20,ED,F6,F0,04,A0,02,D0,C1,20,42,F6, 2770
1086 DATA 4C,63,C2,A6,7A,A0,04,84,0F,BD,00,02,10,07,C9,FF,F0,3E,E8,D0, 2380
1087 DATA F4,C9,20,F0,37,85,08,C9,22,F0,55,24,0F,70,2D,C9,3F,D0,04,A9, 2326
1088 DATA 99,D0,25,C9,30,90,04,C9,3C,90,1D,84,71,A0,00,84,0B,88,86,7A, 2169
1089 DATA CA,C8,E8,BD,00,02,38,F9,9E,A0,F0,F5,C9,80,D0,2F,05,0B,A4,71, 2810
1090 DATA E8,C8,99,FB,01,C9,00,F0,38,38,E9,3A,F0,04,C9,49,D0,02,85,0F, 2557
1091 DATA 38,E9,55,D0,A0,85,08,BD,00,02,F0,E0,C5,08,F0,DC,C8,99,FB,01, 2808
1092 DATA E8,D0,F0,A6,7A,E6,0B,C8,B9,9D,A0,10,FA,B9,9E,A0,D0,B5,F0,0F, 3324
1093 DATA BD,00,02,10,BD,99,FD,01,C6,7B,A9,FF,85,7A,60,A0,00,B9,7E,C7, 2569
1094 DATA D0,02,C8,E8,BD,00,02,38,F9,7E,C7,F0,F5,C9,80,D0,04,05,0B,D0, 2713
1095 DATA 99,A6,7A,E6,0B,C8,B9,7D,C7,10,FA,B9,7E,C7,D0,E0,F0,C6,47,52, 3190
1096 DATA 41,50,48,49,C3,47,43,4C,45,41,D2,47,43,4F,4C,4F,D2,50,4C,4F, 1860
1097 DATA D4,4C,49,4E,C5,54,45,58,D4,43,4F,4C,4F,D2,41,D4,4B,45,D9,4D, 2315
1098 DATA 55,54,C5,56,4F,CC,41,44,53,D2,57,41,56,C5,53,4F,55,4E,C4,53, 2200
1099 DATA 50,4F,49,4E,54,45,D2,53,50,52,49,54,C5,53,45,D4,58,45,58,D0, 2089
1100 DATA 59,45,58,D0,4E,4F,52,4D,D8,4E,4F,52,4D,D9,53,43,4F,4C,4F,D2, 2113
1101 DATA 53,4F,46,C6,47,53,41,56,C5,47,4C,4F,41,C4,44,49,53,CB,4D,55, 2008
1102 DATA 53,49,C3,45,52,52,4F,D2,43,41,54,41,4C,4F,C7,48,41,4C,C6,58, 2007
1103 DATA 59,DA,55,56,D7,4D,4F,56,C5,53,50,4C,4F,D4,48,45,58,A4,42,49, 2194
1104 DATA 4E,A5,54,45,53,54,A8,4A,4F,D9,53,54,45,53,54,A8,00,10,0F,24, 1739
1105 DATA 0F,30,0B,C9,FF,F0,07,C9,CC,B0,06,4C,24,A7,4C,F3,A6,38,E9,CB, 2620
1106 DATA AA,84,49,A0,FF,CA,F0,08,C8,B9,7E,C7,10,FA,30,F5,C8,B9,7E,C7, 3219
1107 DATA 30,05,20,47,AB,D0,F5,4C,EF,A6,A9,7B,A0,C8,8D,08,03,8C,09,03, 2217
1108 DATA A9,A2,A0,C8,8D,0A,03,8C,0B,03,60,20,73,00,20,84,C8,4C,AE,A7, 2023
1109 DATA C9,CC,90,04,C9,EE,90,06,20,79,00,4C,ED,A7,38,E9,CC,0A,AA,BD, 2637
1110 DATA D2,C8,48,BD,D1,C8,48,4C,73,00,A9,00,85,0D,20,73,00,C9,EE,90, 2388
1111 DATA 04,C9,F3,90,06,20,79,00,4C,8D,AE,38,E9,EE,0A,48,20,73,00,68, 2002
1112 DATA A8,B9,15,C9,85,55,B9,16,C9,85,56,20,54,00,4C,8D,AD,37,C2,41, 2240
1113 DATA C2,48,C2,1E,CC,E9,CB,5B,C2,64,C2,82,C2,A0,C2,AC,C2,BA,C2,F4, 3377
1114 DATA C2,13,C3,59,C3,75,C3,97,C3,C3,C3,23,C4,42,C4,61,C4,82,C4,A3, 3015
1115 DATA C4,C4,C4,F1,C4,2E,C5,65,C5,C3,C5,26,C6,45,C6,57,C9,43,CD,44, 3089
1116 DATA CD,E8,C9,80,CC,8D,CA,C9,CA,71,CB,97,CB,CA,CC,A9,43,8D,5F,C1, 3457
1117 DATA 8D,B3,C1,A9,C9,8D,60,C1,8D,B4,C1,20,A9,C0,A9,C9,8D,5F,C1,8D, 3160
1118 DATA B3,C1,A9,C1,8D,60,C1,8D,B4,C1,60,AD,12,CE,85,14,AD,13,CE,85, 2855
1119 DATA 15,AE,22,CE,20,54,C0,51,22,91,22,60,20,79,00,20,9E,B7,8A,F0, 2037
1120 DATA 6D,78,A9,80,8D,14,03,A9,C9,8D,15,03,A9,D9,8D,12,D0,AD,11,D0, 2376
1121 DATA 29,7F,8D,11,D0,A9,81,8D,1A,D0,58,60,AD,19,D0,8D,19,D0,30,07, 2226
1122 DATA AD,0D,DC,58,4C,31,EA,AD,12,D0,C9,FA,B0,1C,AD,11,D0,29,DF,8D, 2710
1123 DATA 11,D0,AD,18,D0,29,F7,8D,18,D0,38,20,2C,CD,A9,FA,8D,12,D0,4C, 2490
1124 DATA BC,FE,AD,11,D0,09,20,8D,11,D0,AD,18,D0,09,08,8D,18,D0,18,20, 2098
1125 DATA 2C,CD,A9,D9,8D,12,D0,4C,BC,FE,78,78,A9,31,8D,14,03,A9,EA,8D, 2686
1126 DATA 15,03,A9,00,8D,12,D0,A9,00,8D,1A,D0,8D,19,D0,58,60,A9,22,8D, 2006
1127 DATA 5F,C1,8D,B3,C1,A9,CA,8D,60,C1,8D,B4,C1,20,79,00,20,9E,B7,8E, 2784
1128 DATA F0,07,20,FD,AE,20,9E,B7,8E,F1,07,20,FD,AE,20,A9,C0,A9,C9,8D, 2832
1129 DATA B3,C1,8D,5F,C1,A9,C1,8D,B4,C1,8D,60,C1,60,AD,12,CE,85,14,AD, 2926
1130 DATA 13,CE,85,15,AD,22,CE,8D,EB,07,AD,F0,07,8D,E9,07,8D,EA,07,20, 2390
1131 DATA E0,C3,A2,19,AC,F1,07,88,D0,FD,CA,D0,F7,60,EA,A9,7B,8D,5F,C1, 3331
1132 DATA 8D,B3,C1,A9,CA,8D,60,C1,8D,B4,C1,20,79,00,20,9E,B7,8E,FF,CE, 2957
1133 DATA 20,FD,AE,20,A9,C0,A9,C9,8D,B3,C1,8D,5F,C1,A9,C1,8D,B4,C1,8D, 3181
1134 DATA 60,C1,60,EA,AC,12,CE,AE,22,CE,18,20,F0,FF,AD,FF,CE,20,D2,FF, 3111
1135 DATA 60,20,BF,CA,20,79,00,4C,99,CA,20,73,00,90,07,C9,41,90,1C,38, 1897
1136 DATA E9,07,38,E9,30,48,A5,61,F0,08,18,18,69,04,B0,0C,85,61,68,F0, 2078
1137 DATA E1,20,7E,BD,4C,96,CA,60,4C,7E,B9,A9,00,A2,0A,95,5D,CA,10,FB, 2535
1138 DATA 60,20,BF,CA,20,79,00,4C,D5,CA,20,73,00,C9,32,B0,E2,C9,30,90, 2358
1139 DATA DE,E9,30,48,A5,61,F0,04,E6,61,F0,D4,68,F0,E7,20,7E,BD,4C,D2, 3068
1140 DATA CA,A9,01,85,2B,A9,40,85,2C,A9,00,8D,00,40,A9,06,8D,20,D0,8D, 2029
```

```
1141 DATA 21,D0,A2,80,EA,EA,EA,A9,1C,A0,CB,20,B0,CD,A9,42,A0,CB,20,B0, 3012
1142 DATA CD,4C,66,CB,08,93,05,11,20,20,20,20,20,20,20,20,20,2A,2A,2A, 1177
1143 DATA 20,4D,55,4C,54,49,20,42,41,53,49,43,20,56,33,20,2A,2A,2A,0D, 1153
1144 DATA 8E,00,11,20,20,20,20,20,20,20,20,20,32,34,35,37,36,20,42,41, 874
1145 DATA 53,49,43,20,42,59,54,45,53,20,46,52,45,45,11,11,0D,00,A9,07, 1191
1146 DATA 8D,02,03,A9,CD,8D,03,03,60,EA,20,4E,C0,85,02,31,22,C5,02,F0, 1956
1147 DATA 05,A2,00,4C,84,CB,A2,01,8A,20,3C,BC,20,79,00,C9,29,F0,05,A2, 1961
1148 DATA 0B,6C,00,03,4C,73,00,A2,00,86,02,4C,B0,CB,0E,01,06,02,07,03, 1099
1149 DATA 05,04,0D,05,09,06,0B,07,0A,08,00,00,AD,00,DC,29,0F,85,02,A0, 822
1150 DATA 00,B9,9E,CB,D0,07,A9,00,85,02,4C,D3,CB,C5,02,F0,05,C8,C8,4C, 2475
1151 DATA B9,CB,B9,9F,CB,85,02,AD,00,DC,29,10,D0,07,A5,02,18,69,64,85, 2264
1152 DATA 02,A5,02,A8,20,A2,B3,4C,79,00,20,79,00,20,9E,B7,E0,00,D0,07, 1872
1153 DATA 20,FD,AE,20,F5,C1,60,E0,01,D0,07,20,FD,AE,20,A9,C0,60,E0,02, 2639
1154 DATA D0,07,20,FD,AE,20,1F,C9,60,E0,03,D0,07,20,FD,AE,20,4A,CA,60, 2339
1155 DATA 4C,64,C2,20,79,00,20,9E,B7,E0,00,D0,07,20,FD,AE,20,44,C0,60, 2182
1156 DATA 20,FD,AE,20,3C,C0,60,20,FD,AE,20,9E,B7,86,FB,86,FD,20,FD,AE, 2902
1157 DATA 20,9E,B7,86,FC,8A,0A,18,65,FC,85,FC,A5,FB,4A,4A,4A,85,FB,A5, 2856
1158 DATA FD,29,07,85,FD,A9,07,38,E5,FD,85,FD,A9,00,18,65,FC,18,65,FB, 2709
1159 DATA 85,FC,A4,FD,F0,0A,A9,01,85,FD,06,FD,88,D0,FB,60,A0,01,84,FD, 3104
1160 DATA 60,20,79,00,20,9E,B7,8A,F0,1F,20,37,CC,20,FD,AE,20,8A,AD,20, 2156
1161 DATA F7,B7,A5,14,65,FC,85,14,90,02,E6,15,A0,00,B1,14,05,FD,91,14, 2298
1162 DATA 60,20,37,CC,20,FD,AE,20,8A,AD,20,F7,B7,A5,14,65,FC,85,14,90, 2486
1163 DATA 02,E6,15,A5,FD,49,FF,A0,00,31,14,91,14,60,20,79,00,20,3A,CC, 1936
1164 DATA 20,FD,AE,20,8A,AD,20,F7,B7,A5,14,65,FC,85,14,90,02,E6,15,A0, 2512
1165 DATA 00,B1,14,25,FD,F0,08,A9,01,20,3C,BC,4C,F8,CC,A9,00,20,3C,BC, 2162
1166 DATA 20,79,00,C9,29,D0,03,4C,73,00,A2,0B,6C,00,03,48,20,46,CD,D0, 1668
1167 DATA 1B,38,20,2C,CD,AD,11,D0,29,DF,8D,11,D0,AD,18,D0,29,F7,8D,18, 2250
1168 DATA D0,A5,B6,F0,03,20,4C,CD,68,4C,83,A4,B0,0B,AD,18,D0,29,0F,09, 2243
1169 DATA 20,8D,18,D0,60,AD,18,D0,29,0F,09,10,8D,18,D0,60,60,60,AD,14, 1841
1170 DATA 03,C9,31,60,A9,06,8D,20,D0,8D,21,D0,A9,01,8D,86,02,A9,00,85, 2036
1171 DATA B6,A9,68,8D,18,03,A9,CD,8D,19,03,60,48,8A,48,98,48,A9,FA,85, 2320
1172 DATA B6,4C,4C,FE,EC,02,CE,8E,22,CE,D0,24,AD,00,CE,8D,12,CE,C5,14, 2619
1173 DATA D0,1A,AD,01,CE,8D,13,CE,C5,15,D0,10,AD,5F,C1,AC,60,C1,8D,9D, 2642
1174 DATA CD,8C,9E,CD,20,FF,FF,60,4C,CF,C0,A2,08,CA,BD,F8,07,9D,F8,0B, 3053
1175 DATA 8A,D0,F6,60,85,14,84,15,A0,00,B1,14,F0,07,20,D2,FF,C8,4C,B6, 2553
1176 DATA CD,60,55, 386
```

Multi-Basic Befehlsliste

Graphik-Befehle:

GRAPHIC — schaltet auf HGR um

TEXT — schaltet wieder zurück auf Textmodus und löscht den Bildschirm

GCOLOR hf+16*pf — definiert die Farben mit denen in HGR gezeichnet wird:
hf = Hintergrundfarbe
pf = Punktfarbe

GCLEAR — löscht den HGR-Bildschirm

PLOT m,x,y — setzt einen Punkt in den mit GCOLOR definierten Farben.
m = Zeichenmodus (0=löschen/ 1=setzen)
x = X-Koordinate (0-319)
y = Y-Koordinate (0-199)

LINE m,ax,ay,ex,ey — zieht eine Linie
m = Zeichenmodus
0 ≙ Linie löschen
1 ≙ Linie ziehen
2 ≙ invertieren
(Punkt gesetzt = löschen und umgek.)
3 ≙ siehe LINE 3, ax,...
ax + ay = Koord. Startpunkt
ex + ey = Koord. Endpunkt

LINE 3,asc,ax,ay,ex,ey — zieht eine Linie im Textmodus
asc = ASCII Zeichen mit dem gezeichnet wird (siehe HB)
z. B. a = 64

ax(0-39), ay(0-24) — Koord. Startpunkt

ex(0-39), ey(0-24) — Koord. Endpunkt

TEST(x,y) — testet ob bei Koord. x,y ein Punkt gesetzt ist.
1 = Punkt gesetzt
0 = kein Punkt gesetzt
z. B. PRINT TEST(57,136)

HALF m — teilt den Bildschirm: oben HGR-Anzeig
unten vier Textzeilen
m = 1 Teilung ein
m = 0 Teilung aus
Vor Disketten- oder Kassettenoperationen sollte die Teilung ausgesch. werden!

STEST(x,y,ad) — testet ob in dem Sprite ab Adresse ad ein Punkt gesetzt ist (1) oder nicht (0)
z. B. PRINT TEST (1,10,704)
(704 = 64*11 = Block 11)

Musik-Befehle:

VOL n — definiert die Lautstärke für alle Stimmen
n = 0-15)

MUTE — löscht den gesamten Soundchip (RESET)

ADSR s,y*16+d,s*16+r — definiert die Hüllkurve für Stimme s (1-3)
a(ttack),d(ecay),s(ustain),r(elease), jeweils 0-15 bitte näheres im Handbuch nachschlagen!

WAVE s,f — setzt die Wellenform für Stimme s
f : 0 = Stimme klingt aus
1 = Dreieck
2 = Sägezahn
3 = Rechteck
4 = Rauschen

SOUND s,fr — gibt Stimme s die Tonhöhe f (0-65535)
siehe Tabelle im Handbuch!

MUSIC s,ok,"N" — spielt auf Stimme s die Note "N" in Oktave ok (0-6)
N: c,d,e,f,g,a,h erhöhte Noten mit SHIFT (#C = shift C)
Es wird immer nur eine Note gespielt!

Sonstige Befehle:

COLOR rf,hf,zf — setzt Bildschirmfarben: rf = Rahmenf.
hf = Hintergrd.
zf = Schriftf.

KEY — wartet bis irgendeine Taste gedrückt wird

AT x,y,"..." — setzt Bildschirmausgabe an Position x (0-39) und y (024). Nach den Koord. fährt der Befehl wie PRINT fort.
z. B. AT 10,5,A$
AT 20,11,"HALLO"; usw.

JOY — ist eine Variable die den aktuellen Wert des Joysticks an Port 2 enthält:
812
703 bei gedr. Feuerknopf
654 wird 100 addiert

HEX$ ffff — rechnet die max. 4-stellige Hexadezimalzahl nach dezimal um
z. B. PRINT HEX$ AEFD

BIN% bbbbbbbb — rechnet die max. 8-stellige Binärzahl bbbbbbbb nach dezimal um
z. B. PRINT BIN% 10101101

GLOAD "name",g — lädt einen HGR-Bild mit Namen "name" von Gerät Nummer g (1=Kas./8=Diskette)

GSAVE "name",g — speichert einen HGR-Bildschirm unter dem Namen "name" auf Gerät Nummer g ab (1=Kas./8=Diskette)

Disk-Befehle:

CATALOG — zeigt das Inhaltsverzeichnis einer Diskette auf dem Bildschirm an, ohne das Programm im Speicher zu zerstören

ERROR — zeigt den Fehlerkanal der Floppy auf dem Bildschirm an

DISK "befehl" — sendet einen Befehl zur Floppy
z. B. DISK "N:TEST,00" zum Formatieren einer Diskette

Sprite-Befehle:

SPRITE n — schaltet Sprite Numme n (0-7) ein

SOFF — schaltet alle Sprites aus

SOFF n — schaltet Sprite Nummer n aus

SPOINTER n,b — richtet den Zeiger von Sprite Nummer n auf Datenblock b (0-255) siehe Handbuch!

SCOLOR n,f — gibt dem Sprite n die Farbe f (0-15)

XEXP n) — dehnt Sprite n in X- bzw. Y-Richtung aus

YEXP n)

NORMX n) — gibt Sprite n in X- bzw. Y-Richtung die normale Größe

NORMY n)

SET n,x,y — setzt Sprite n auf Koord. x (0-512) und y (0-255)

MOVE n,sp,ax,ay,ex,ey — bewegt Sprite n von Position ax, ay mit Geschwindigkeit sp (1-255) nach ex,ey

SPLOT m,x,y,ad — setzt einen Punkt in einem Sprite
m = Modus : 0=löschen / 1=setzen
x = X-Koord. Punkt (0-23)
y = Y-Koord. Punkt (0-27)
ad = Adresse des Sprites!
Adresse = Block*64 !!!

Commodore 64: „BILDEDITOR"

Komfortables Zeichenprogramm mit integriertem Programmgenerator

Dies Programm ist nur auf Diskette lauffähig

Der Grund, dieses Programm zu schreiben, war meine Verärgerung über die Probleme bei der Erstellung von Bildschirmgrafiken im Textmodus. Da die Grafikzeichen des C-64 zum Erstellen von Hintergrundbildern für Spiele fast immer ungeeignet sind, wurde in den 'BILDEDITOR' gleich noch ein Zeicheneditor mit eingebaut, um die benötigten Zeichen (z. B. für Wolken, Landschaft usw.) selbst kreieren zu können. Wird der 'BILDEDITOR' mit 'RUN' gestartet, so muß man zuerst die Zeilennummer eingeben, mit der das später erzeugte Programm beginnt.

Dannach meldet sich der 'BILDEDITOR' mittels eines Eingabecursors. Der Cursor kann ganz normal mit den Cursortasten bewegt werden. Die Farbwahl erfolgt ebenfalls wie gewöhnlich mit 'C= bzw. Ctrl' + der entsprechenden Farbtaste. Zeichen werden nach dem Drücken von 'Ctrl+RVON' revers dargestellt und erscheinen

nach dem Betätigen von 'Ctrl+ RVOFF' wieder normal auf dem Bildschirm.
Hat man nun auf dem Bildschirm die Grafik fertiggestellt, so wird nach Betätigen der Taste 'F7' das Bild in 'PRINT'-Zeilen übernommen. Dabei werden die verschiedenen Farben, sowie die reversen Zeichen durch entsprechende Steuerzeichen innerhalb der 'PRINT'-Anweisung erreicht. Das auf diese Weise erhaltene Programm kann nun abgespeichert, weiter editiert oder in eigenen Programmen verwendet werden.

Mit dem 'BILDEDITOR' lassen sich nur die obersten 24 der insgesamt 25 Bildschirmzeilen bearbeiten. Der Grund hierfür ist, daß sich, bei einer 40 Zeichen langen 'PRINT'-Anweisung in der untersten Zeile, der gesamte Bildschirminhalt nach oben verschiebt.
Es bliebe nur die Möglichkeit die ersten 39 Zeichen zu PRINT'en und das letzte Zeichen rechts unten mittels 'POKE' ins Video-, sowie Farbram zu setzen, was mir aber etwas zu umständlich erschien.
Nach Tastendruck auf 'F5' befindet man sich im Zeicheneditor. Nun kann man sich eventuell benötigte Zeichen definieren oder einen Zeichensatz laden bzw. abspeichern. Durch Druck auf 'E' gelangt man wieder in den Bildeditor, wo man mit seiner Arbeit fortfahren kann.

Noch ein paar Worte zum Zeichensatz. Die neuen Zeichen stehen im Speicher ab 12288.
Ein Zeichensatz kann mit 'LOAD "NAME",8,1' von Diskette eingeladen und mit 'POKE 53272,29' eingeschaltet werden. Ein Demo-Zeichensatz befindet sich bereits auf der Diskette. Im folgenden werden sämtliche Befehle von Bildeditor und Zeicheneditor aufgeführt.

BILDEDITOR

F7 Wahl der Rahmenfarbe
F2 Wahl der Hintergrundfarbe
F5 Sprung zum Zeicheneditor. Der Bildschirminhalt geht nicht verloren!
F7 Übernahme des Bildtextes in BASIC-Zeilen. Nach Ausführung dieses Programmteils ist der 'BILDEDITOR' im Speicher gelöscht. Es bleiben nur die 'PRINT'-Zeilen für die Grafik übrig.

ZEICHENEDITOR

A Befehlsübersicht des Zeicheneditors
F1 Rahmenfarbe
F3 Hintergrundfarbe
H Zeichen holen. Auf dem Bildschirm erscheint die Zeichenmatrix des gewählten Zeichens.
Clr Zeichenmatrix löschen
I Zeichen invertieren
C Copy. Die Zeichenmatrix aus dem linken Feld wird ins rechte Feld kopiert.
Crsr Cursorsteuerung im Eingabefeld
+ Setzen eines Punktes an die Cursorposition
- Löschen eines Punktes unter dem Cursor
F Farben ändern
Farbe 1 = Zeichenfarbe
Farbe 2 = Multicolorfarbe 1
Farbe 3 = Multicolorfarbe 2
Bei eingeschaltetem Multicolormodus kann ein Zeichen max. 4-farbig sein. Farbe 1 muß dann größer sein als 7.
ß Ein- bzw. Ausschalten des Multicolormodus
D Definieren eines Zeichens

Zum Schluß noch einige Bemerkungen zum Abtippen und Abspeichern.
Zunächst wird Teil 1 des Listings abgetippt und abgespeichert. Dabei ist zu beachten daß keine zusätzlichen Programmzeilen eingegeben werden, da später die Maschinenroutinen unmittelbar hinter dem Basicteil stehen.
Jetzt wird Teil 2 des Listings eingegeben, wobei darauf zu achten ist, daß die gleichen Zeilennummern verwendet werden. Dies ist wichtig, da bei einem Prüfsummenfehler die Zeilennummer der fehlerhaften Datazeile angegeben wird.
Beim Starten von Teil 2 muß das Diskettenlaufwerk eingeschaltet sein, da nun ein Maschinenprogramm mit dem Namen 'DATA' auf die Diskette geschrieben wird.
Zum Schluß wird zunächst Teil 1 geladen, danach lädt man mit 'LOAD"DATA",8,1' das Maschinenprogramm nach und speichert beides mit 'SAVE" BILDEDITOR",8' ab.

Jetzt kann das Programm gestartet werden. Sollten sich irgendwo Tippfehler eingeschlichen haben, so muß nach der Korrektur wieder 'DATA' nachgeladen werden. Dies ist leider notwendig, da sich beim Ändern des BASIC-Textes die Maschinenroutinen verschieben und das Programm nach dem Starten abstürzen würde.

BILDEDITOR TEIL 1

```
100 REM ************************
110 REM **      BILDEDITOR      **
120 REM **  1984 BY K.SELIGER   **
130 REM **  AM PFINGSTBORN 41   **
140 REM **  6760 ROCKENHAUSEN   **
150 REM **     06361/7649       **
160 REM ************************
170 REM ZEILENNUMMER
180 PRINTCHR$(147)
190 INPUT"START ZEILENNUMMER (0-63975)";ZN
200 IFZN>63975ORZN<0THENPRINT"FALSCHE WERTE":GOTO190
210 POKE831,INT(ZN/256):POKE830,ZN-256*PEEK(831)
220 REM INITIALISIEREN
230 POKE45,0:POKE46,80:RUN240
240 V=53248:MO=1:FORA=0TO7:A%(7-A)=2^A:NEXT
250 POKEV+21,1:POKEV+39,1:POKE2040,133
260 RE=8987:VE=8939:DI=9006:F1=4
270 SYSVE
280 POKE2,16:POKE3,0:POKE4,48:POKE5,0:POKE6,192:SYSRE
290 REM BILDEDITOR
```



```
300 REM **********
310 PRINTCHR$(147);
320 PRINTCHR$(9);
330 GETA$
340 IFA$=CHR$(133)THENF1=F1+1:POKE53280,(F1AND255)
350 IFA$=CHR$(134)THENF2=F2+1:POKE53281,(F2AND255)
360 IFA$=CHR$(136)THENPOKEV+21,0:SYS8576
370 IFA$=CHR$(135)THEN450:REM ZEICHENEDITOR
380 HY=PEEK(214)
390 HX=PEEK(211):IFHX>39THENHX=HX-40
400 HJ=HX*8+24:HH=INT(HJ/256):HL=HJ-256*HH
410 POKEV,HL:POKEV+16,HH:POKEV+1,HY*8+50
420 IFHY=23ANDA$=CHR$(17)THEN330
430 IFHY>23THENPRINTCHR$(145);:GOTO330
440 PRINTA$;:GOTO330
450 REM
460 REM ZEICHENEDITOR
470 REM **************
480 REM
490 REM BILD RETTEN
500 POKE2,4:POKE3,0:POKE4,4:POKE5,0:POKE6,40:SYSRE
510 POKE2,4:POKE3,0:POKE4,216:POKE5,0:POKE6,44:SYSRE
520 SYSVE
530 POKEV,X*8+144:POKEV+1,Y*8+98:POKEV+16,0
540 PRINTCHR$(142);CHR$(8)
550 REM EINGABEFELD
560 POKEV+21,1
570 PRINTCHR$(147);
580 PRINTCHR$(156);CHR$(146)
590 PRINT"┌──────────────────────────────────────┐";
600 PRINT"│              ZEICHENEDITOR           │";
610 PRINT"└──────────────────────────────────────┘";
620 PRINT"                                        ";
630 PRINT"┌──────────┐ ┌──────────┐ ┌──────────┐";
640 PRINT"│ ┌──────┐ │ │ ┌──────┐ │ │ ANLEITUNG: │";
650 PRINT"│ │      │ │ │ │      │ │ │           │";
660 PRINT"│ │      │ │ │ │      │ │ │ TASTE 'A' │";
670 PRINT"│ │      │ │ │ │      │ │ │           │";
680 PRINT"│ │      │ │ │ │      │ │ └───────────┘";
690 PRINT"│ │      │ │ │ │      │ │ ┌───────────┐";
700 PRINT"│ │      │ │ │ │      │ │ │           │";
710 PRINT"│ └──────┘ │ │ └──────┘ │ │"CHR$(169)" MULTI "MM" │";
720 PRINT"└──────────┘ └──────────┘ │           │";
730 PRINT"                         │ MODUS "MO" │";
740 PRINT"ZEICHEN ALT  ZEICHEN NEU └───────────┘";
750 PRINT"                                        ";
760 PRINT"└──────────────────────────────────────┘";
```

```
770 POKE55842,F3:GOTO1160
780 REM TASTATURABFRAGE
790 GETA$
800 IFA$=CHR$(17)THEN1020
810 IFA$=CHR$(145)THEN1050
820 IFA$=CHR$(29) THEN1080
830 IFA$=CHR$(157)THEN1110
840 IFA$=CHR$(147)THEN1160
850 IFA$="H"THEN1200
860 IFA$="I"THEN1380
870 IFA$="+"THEN1410
880 IFA$="-"THEN1450
```

```
890 IFA$="L"THEN1490
900 IFA$="S"THEN1570
910 IFA$="C"THEN1830
920 IFA$="D"THEN1650
930 IFA$="M"THEN1750
940 IFA$="A"THEN2180
950 IFA$="E"THEN2110
960 IFA$="$"THEN1860
970 IFA$="F"THEN1890
980 IFA$="^"THEN1960
```

```
990 IFA$=CHR$(133)THENF1=F1+1:POKE53280,(F1AND255)
1000 IFA$=CHR$(134)THENF2=F2+2:POKE53281,(F2AND255)
1010 GOTO790
1020 REM CURSOR RUNTER
1030 Y=Y+1:IFY>7THENY=0
1040 GOTO1130
1050 REM CURSOR HOCH
1060 Y=Y-1:IFY<0THENY=7
1070 GOTO1130
1080 REM CURSOR RECHTS
1090 X=X+1:IFX>7THENX=0
1100 GOTO1130
1110 REM CURSOR LINKS
1120 X=X-1:IFX<0THENX=7
1130 REM CURSOR BEWEGEN
1140 POKEV,X*8+144:POKEV+1,Y*8+98
1150 GOTO790
1160 REM LOESCHEN
1170 FORA=0TO7:POKE12288+105*8+A,0:NEXT
1180 FORA=0TO7:FORB=0TO7:POKE1279+40*A+B,79:NEXTB:NEXTA
1190 GOTO790
1200 REM ZEICHEN HOLEN
1210 GOSUB2020
1220 PRINTSPC(13)"ZEICHEN HOLEN"
1230 PRINT"  WELCHES ZEICHEN":POKE198,0:WAIT198,1:GETZE$
1240 ZB=ASC(ZE$)
1250 IFZB>191THENZB=ZB-128:GOTO1270
1260 IFZB>63 THENZB=ZB-64
1270 FORA=0TO7:POKE12288+105*8+A,PEEK(49152+ZB*8+A+M1):NEXT
1280 FORA=0TO7:FORB=0TO7
1290 SE=PEEK(12288+8*ZB+A+M1):SF=PEEK(12288+8*105+A)
1300 IF(SEANDA%(B))THENPOKE1267+40*A+B,160:GOTO1320
1310 POKE1267+40*A+B,79
1320 IF(SFANDA%(B))THENPOKE1279+40*A+B,160:GOTO1340
1330 POKE1279+40*A+B,79
1340 NEXTB:NEXTA
1350 PRINTCHR$(19);:FORA=1TO7:PRINT:NEXT:POKE646,PEEK(55842)
1360 PRINTSPC(26)CHR$(169)CHR$(156)
1370 GOSUB2020:GOSUB2040:GOTO790
1380 REM INVERTIEREN
1390 FORA=0TO7:POKE12288+8*105+A,255-PEEK(12288+8*105+A):NEXT
1400 GOTO1280
1410 REM PUNKT SETZEN
1420 POKE 1279+X+40*Y,160
1430 POKE12288+105*8+Y,PEEK(12288+105*8+Y)OR2^(7-X)
1440 GOTO790
1450 REM PUNKT LOESCHEN
1460 POKE1279+X+40*Y,79
1470 POKE12288+105*8+Y,PEEK(12288+105*8+Y)AND255-(2^(7-X))
1480 GOTO790
1490 REM LADEN VON ZEICHENSAETZEN
1500 POKEV+21,0:POKEV+24,21
1510 GOSUB2020
1520 PRINTSPC(7)"LADEN VON ZEICHENSAETZEN"
1530 INPUT"  FILENAME";NA$
1540 SYS57812 NA$,8,1:POKE780,0:SYS65493
1550 POKE2,16:POKE3,0:POKE4,48:POKE5,0:POKE6,192:SYSRE:SYSVE
1560 GOSUB2050:GOSUB2020:GOSUB2040:GOTO550
1570 REM ZEICHENSATZ ABSPEICHERN
1580 POKEV+24,21:POKEV+21,0:POKE2,16:POKE3,0:POKE4,192:POKE5,0:POKE6,48:SYSRE
```



```
1590 GOSUB2020
1600 PRINTTAB(8)"ZEICHENSATZ ABSPEICHERN"
1610 INPUT"  FILENAME";NA$:NA$="@:"+NA$
1620 SYS57812 NA$,8
1630 POKE193,0:POKE194,48:POKE175,64:SYS62954
1640 GOSUB2050:GOSUB2020:GOSUB2040:SYSVE:GOTO550
1650 REM ZEICHEN DEFINIEREN
1660 GOSUB2020
1670 PRINTTAB(9)"ZEICHEN DEFINIEREN"
1680 PRINT"  TASTE DRUECKEN"
1690 POKE198,0:WAIT198,1:GETZE$
1700 ZB=ASC(ZE$)
1710 IFZB>191THENZB=ZB-128:GOTO1730
1720 IFZB>63 THENZB=ZB-64
1730 FORA=0TO7:POKE49152+ZB*8+M1+A,PEEK(12288+105*8+A):NEXT
1740 GOSUB2020:GOSUB2040:GOTO790
1750 REM MODUSWECHSEL
1760 GOSUB2020
1770 PRINTSPC(11)"MODUSWECHSEL"
1780 PRINT"  WELCHER ZEICHSATZ(1-4)"
1790 POKE198,0:WAIT198,1:GETMO$:MO=VAL(MO$):IFMO<1ORMO>4THEN1790
1800 M1=MO*1024-1024
1810 PRINTCHR$(19):FORA=0TO6:PRINT:NEXT:PRINTSPC(35)MO
1820 GOSUB2020:GOSUB2040:GOTO790
1830 REM ZEICHEN COPIEREN
1840 FOR A=0TO7:POKE49152+8*ZB+A+M1,PEEK(12288+8*ZB+A+M1):NEXT
1850 GOTO1270
1860 REM DIREKTORY LADEN
1870 PRINTCHR$(147):POKEV+21,0:SYSDI
1880 POKE198,0:WAIT198,1:GOTO550
1890 REM MULTICOLORFARBEN
1900 GOSUB2020:PRINTSPC(11)"MULTICOLORFARBEN"
1910 INPUT"  FARBE 1 (0-15)";F3$:F3=VAL(F3$):POKE55842,(F3AND15)
1920 INPUT"  FARBE 2 (0-15)";F4$:F4=VAL(F4$):POKEV+34,(F4AND15)
1930 INPUT"  FARBE 3 (0-15)";F5$:F5=VAL(F5$):POKEV+35,(F5AND15)
1940 GOSUB2020:GOSUB2040
1950 GOTO790
1960 REM MULTICOLOR EIN/AUS
1970 MM=MM+1:IFMM=2THENMM=0
1980 POKEV+22,PEEK(V+22)AND239
1990 IFMM=1THENPOKEV+22,PEEK(V+22)OR16
2000 PRINTCHR$(19);:FORA=1TO7:PRINT:NEXT:PRINTSPC(35)MM:GOTO790
2010 REM POSITIONIEREN
2020 PRINTCHR$(19);:FORA=1TO10:PRINT:NEXT
2030 RETURN
2040 FORA=1TO4:PRINT"                                                    ":NEXT:RETURN
2050 REM FEHLERKANAL
2060 OPEN1,8,15
2070 INPUT#1,FE,FE$:CLOSE1
2080 PRINTSPC(5)FE;SPC(5)FE$
2090 POKE198,0:WAIT198,1
2100 RETURN
2110 REM
2120 REM ZUM BILDEDITOR
2130 REM --------------
2140 POKE2,4:POKE3,0:POKE4,40:POKE5,0:POKE6,4:SYSRE
2150 POKE2,4:POKE3,0:POKE4,44:POKE5,0:POKE6,216:SYSRE
2160 POKE2,16:POKE3,0:POKE4,192:POKE5,0:POKE6,48:SYSRE
2170 GOTO320
2180 REM ANLEITUNG
```

```
2190 REM ---------
2200 POKEV+21,0:PRINTCHR$(147)
2210 PRINTSPC(12)"BEFEHLSSATZ"
2220 PRINT" ---------------------------------------"
2230 PRINT"    H  :  ZEICHEN HOLEN"
2240 PRINT"    I  :  ZEICHEN INVERTIEREN"
2250 PRINT"    D  :  ZEICHEN DEFINIEREN"
2260 PRINT"    M  :  MODUS WECHSELN"
2270 PRINT"    +  :  PUNKT SETZEN"
2280 PRINT"    -  :  PUNKT LOESCHEN"
2290 PRINT"    L  :  ZEICHENSATZ LADEN"
2300 PRINT"    S  :  ZEICHENSATZ ABSPEICHERN"
2310 PRINT"    E  :  ZUM BILDEDITOR"
2320 PRINT"    $  :  DIRECTORY LADEN"
2330 PRINT"    C  :  ZEICHEN KOPIEREN"
2340 PRINT"    CLR:  ZEICHEN LOESCHEN"
2350 PRINT"    F  :  MULTICOLOR FARBEN"
2360 PRINT"    ^  :  MULTICOLOR EIN/AUS"
2370 PRINT"    F1 :  RAHMENFARBE"
2380 PRINT"    F3 :  BILDSCHIRMFARBE"
2390 PRINT"    CURSORTASTEN : STEUERUNG"
2400 POKE198,0:WAIT198,1:POKE198,0:GOTO550
READY.
```

Die Antwort auf diese Frage finden Sie auf Seite 91.

```
0 REM ************************
1 REM ** BILDEDITOR  TEIL 2 **
2 REM **  MC-CODE WIRD AUF  **
3 REM **  DISK GESCHRIEBEN  **
4 REM ************************
10 ZE=2000:BA=8512:NA$="@:DATA"
20 FORA=1TO63:SU=0
30 FORB=0TO8:READDA:POKEBA+B,DA:SU=SU+DA:NEXTB
40 READSD:IFSD<>SUTHENPRINT"DATAFEHLER IN ZEILE"ZE:END
50 ZE=ZE+1:BA=BA+9
60 NEXTA
70 SYS57812 NA$,8
80 POKE193,64:POKE194,33
90 POKE174,118:POKE175,35
100 SYS62954
110 :
2000 DATA255,0,0,129,0,0,129,0,0, 513
2001 DATA129,0,0,129,0,0,129,0,0, 387
2002 DATA129,0,0,255,0,0,0,0,0, 384
2003 DATA0,0,0,0,0,0,0,0,0, 0
2004 DATA0,0,0,0,0,0,0,0,0, 0
2005 DATA0,0,0,0,0,0,0,0,0, 0
2006 DATA0,0,0,0,0,0,0,0,0, 0
2007 DATA0,169,1,133,78,133,69,169,8, 760
2008 DATA133,79,133,70,169,0,133,3,169, 889
2009 DATA4,133,4,169,255,133,5,169,215, 1087
2010 DATA133,6,169,24,141,64,3,160,0, 700
2011 DATA165,69,24,105,2,133,78,165,70, 811
2012 DATA105,0,133,79,173,62,3,145,78, 778
2013 DATA32,205,34,173,63,3,145,78,32, 765
2014 DATA205,34,169,153,145,78,32,205,34, 1055
2015 DATA169,34,145,78,32,205,34,32,123, 852
2016 DATA34,177,3,201,128,144,6,32,165, 890
2017 DATA34,76,231,33,174,65,3,224,1, 841
2018 DATA208,3,32,190,34,201,32,176,6, 882
2019 DATA24,105,64,76,2,34,201,96,144, 746
```

BILDEDITOR TEIL 2


```
2020 DATA6,24,105,64,76,2,34,201,64, 576
2021 DATA144,3,24,105,32,145,78,200,192, 923
2022 DATA40,208,197,169,34,145,78,200,169, 1240
2023 DATA59,145,78,200,169,0,145,78,200, 1074
2024 DATA152,160,0,24,101,78,133,78,145, 871
2025 DATA69,170,200,165,79,105,0,133,79, 1000
2026 DATA145,69,133,70,138,133,69,165,3, 925
2027 DATA24,105,40,133,3,165,4,105,0, 579
2028 DATA133,4,165,5,24,105,40,133,5, 614
2029 DATA165,6,105,0,133,6,173,62,3, 653
2030 DATA24,105,1,141,62,3,173,63,3, 575
2031 DATA105,0,141,63,3,206,64,3,240, 825
2032 DATA3,76,161,33,169,0,168,145,78, 833
2033 DATA32,205,34,152,145,78,32,205,34, 917
2034 DATA165,78,133,45,165,79,133,46,00, 844
2035 DATA177,5,41,15,141,66,3,200,177, 825
2036 DATA5,41,15,170,136,205,66,3,240, 881
2037 DATA13,177,3,201,32,208,8,173,66, 881
2038 DATA3,200,145,5,136,96,189,219,34, 1027
2039 DATA145,78,32,205,34,96,56,233,128, 1007
2040 DATA174,65,3,224,1,240,14,162,1, 884
2041 DATA142,65,3,170,169,18,145,78,32, 822
2042 DATA205,34,138,96,162,0,142,65,3, 845
2043 DATA170,169,146,145,78,32,205,34,138, 1117
2044 DATA96,165,78,24,105,1,133,78,165, 845
2045 DATA79,105,0,133,79,96,144,5,28, 669
2046 DATA159,156,30,31,158,129,149,150,151, 1113
2047 DATA152,153,154,155,120,165,1,72,41, 1013
2048 DATA251,133,1,169,208,133,3,169,48, 1115
2049 DATA133,5,160,0,132,2,132,4,162, 730
2050 DATA16,177,2,145,4,200,208,249,230, 1231
2051 DATA3,230,5,202,208,242,104,133,1, 1128
2052 DATA88,169,29,141,24,208,96,166,2, 923
2053 DATA160,0,177,3,145,5,200,208,249, 1147
2054 DATA230,4,230,6,202,208,240,96,169, 1385
2055 DATA36,133,2,169,1,162,2,160,0, 665
2056 DATA32,249,253,169,2,166,186,160,0, 1217
2057 DATA32,0,254,169,0,162,0,160,64, 841
2058 DATA134,95,132,96,32,213,255,165,95, 1217
2059 DATA164,96,32,55,165,173,0,3,72, 760
2060 DATA173,1,3,72,169,61,141,0,3, 623
2061 DATA169,227,141,1,3,32,195,166,104, 1038
2062 DATA141,1,3,104,141,0,3,96,0, 489
```

Eine Basic-Erweiterung für den Schneider CPC 464

Der Schneider CPC 464 besitzt einen sehr komfortablen Basic-Interpreter mit einem riesigen Befehlsumfang. Wer jedoch in Basic maschinennah programmieren will, stößt recht bald an die Grenzen des Locomotive-Basic. Das Programm MACHINE KIT erweitert das Basic um sechs neue RSX-Befehle, die z. B. den Zugriff auf alle ROM- und RAM-Bereiche, die direkte Programmierung des Video-Chips 6845 und des Soundchips AY-3-8912 und den direkten Aufruf von Maschinenroutinen überall im Speicher gestatten. Außerdem erfahren Sie in diesem Beitrag einiges über Maschinenprogrammierung auf dem CPC 464.

Das MACHINE KIT ist selbstverständlich in Maschinensprache geschrieben. Sie geben den Basic-Loader ein. Dieser initialisiert die Befehlserweiterung mit CALL **init,** wobei **init** für die gewählte Startadresse des Programms steht, im vorliegenden Quellcode für 41783.

RSX-Befehle werden, sobald sie ins Betriebssystem integriert sind, wie Basic-Befehle verwendet und sind nur am vorangehenden senkrechten Strich (|) erkenntlich. Solange das Maschinenprogramm selbst nicht gelöscht wird, bleiben sie jederzeit verfügbar – im Direktmodus genauso wie in Programmen.

Das MACHINE KIT umfaßt folgende neue Befehle:

IVIDEO.REG,Register, Wert
Hiermit können Sie die Register des Video-Chips direkt ändern. Auf diese Art können Sie ein Softscrolling programmieren oder auch z. B. mehr als 80 Zeichen pro Zeile darstellen. **Tabelle 1** zeigt eine komplette Liste der Register des 6845-Videochips, hier einige Beispiele dafür:
- Die Video-Register 6 und 7 sind laut Firmware-Handbuch für die Zahl der dargestellten Bildschirmzeilen **(Vertical Displayed)** und den Zeitpunkt des Impulses für die vertikale Synchronisation **(VSync Position)** zuständig. Das heißt, daß Änderungen dieses Registerwertes das Bild auf dem Monitor verschieben. Mit folgenden zwei Befehlen können Sie so z. B. das Monitorbild erheblich vergrößern:

IVIDEO.REG,7,35:IVIDEO.REG,6,30

Sie werden bemerken, daß die obersten Zeilen auf dem Monitor am unteren Bildschirmende wieder auftauchen. Dieses Phänomen ist ganz einfach zu erklären: Um das erweiterte Bild im Speicher unterzubringen, müßte der Bildschirmspeicher größer als die üblichen 16 KByte sein. Da der Video-Chip aber nur über 14 Adreßleitungen mit dem RAM verbunden ist, wird so ein Teil des Speichers doppelt angesprochen und erscheint mehrfach auf dem Bildschirm. Falls dieser Effekt in Programmen sogar erwünscht ist, kann man sich bei entsprechender Programmierung damit die Bildschirmfläche vergrößern.
- Auch ein Softscrolling des Bildschirms ist durch Veränderung der Registerwerte möglich. Dafür zuständig ist das Video-Register 5, das hier für eigene Zwecke „mißbraucht" wird. Schneider bezeichnet das Register als **„Vertical Total Adjust".** Das Register wird normalerweise nur beim Systemstart benötigt und erlaubt einen genauen Abgleich der Bildwiederholfrequenz. Durch Veränderung des Registerwertes läßt sich das gesamte Bild punktweise nach oben oder unten schieben. Sinnvoll ist eigentlich nur die Programmierung in Maschinensprache, für Testzwecke und Versuche reicht Basic aber allemal. Mit den folgenden Befehlen können Sie den Bildschirm vierzig mal um insgesamt eine Buchstabenposition nach oben und unten schieben lassen:

```
FOR l=1 TO 40:FOR j=1 TO 8:IVIDEO.REG,5,j:FOR
k=1 TO 20:NEXT k,j:
FOR j=7 TO 2 STEP -1:IVIDEO.REG,5,j:FOR k=1
TO 20:NEXT k,j,l
```

ISOUND.REG,Register,Wert
Hier ist eine direkte Programmierung ungleich schwieriger als beim Video-Chip, da die vierzehn Register des AY-3-8912-Chips von General Instruments als Multifunktions-Register ausgelegt sind. Wer den Tonchip aber direkt programmiert, kann mit einer höheren Ausführungsgeschwindigkeit rechnen als bei Benutzung der diversen ROM-Routinen, da dem Computer ja ein Großteil der Rechenarbeit abgenommen wird. Die **Tabelle 2** zeigt die genaue Registerbelegung, hier nur ein einfaches Beispiel: Geben Sie zuerst SOUND 1,1000,100 ein und merken Sie sich den Ton. Zum Unterschied geben Sie dann ein: SOUND: 1,1000,100:ISOUND.-REG,7,0. Wie Sie eindeutig hören können, wird der Ton mit Rauschen unterlegt. Aus der Tabelle 2 können Sie die einzelnen Funktionen des Registers 7 ersehen. Durch Setzen aller Bits auf Null werden alle Rauschgeneratoren eingeschaltet und erzeugen den merkwürdigen Ton. Mit den Bits 6 und 7 des Registers 7 hat es eine besondere Bewandtnis: Da der Tonchip einen eingebauten Parallel-Datenport besitzt, wird im CPC 464 zur Kosten- und Bauteileinsparung ein Teil der Tastatur indirekt über den Sound-Chip abgefragt! Die beiden Bits bestimmen, ob die Ports A und B (von denen in dieser Version des Tonchips nur der A-Port benutzt werden kann!) als Ein- oder Ausgang verwendet werden sollen. Damit sollte das Bit 6 dieses Registers immer gelöscht sein, da sonst von der Tastatur wirre Tastenkombinationen angenommen und auf dem Bildschirm ausgegeben werden.

IDOKE,Adresse,Wert
Dies ist das 16-Bit-Äquivalent zum POKE-Befehl, da mit DOKE zwei Bytes bequem auf einmal gePOKEt werden können. Anwendungsfälle sind z. B. zu verändernde Sprünge in Maschinenprogrammen oder Pointer des Basic-Interpreters, z. B. wenn der Basic-Programmstart verschoben werden soll, um Platz für Maschinenprogramme zu schaffen. Mit dem folgenden DOKE-Befehl kann die Baud-Rate, mit der Daten zum eingebauten Cassettenrecorder übertragen werden, auf 3600 Baud eingestellt werden. Dies ist eine der höchsten gerade noch möglichen Übertragungsgeschwindigkeiten und sollte nur bei gutem Bandmaterial verwendet werden:

IDOKE,&B8D1,7942

IROMPEEK,Adresse, Variable
Aufgrund der Hardware-Konzeption des CPC 464 überlagert sich im Bereich &0000 bis &3FFF das Betriebssystem-ROM mit RAM und im Bereich &C000 bis &FFFF das Basic-ROM mit dem Bildschirm-RAM. Der normale PEEK-Befehl spricht immer nur den RAM-Bereich an, ROMs zu PEEKen ist nur in Maschinensprache möglich. Der ROMPEEK-Befehl gestattet es nun auch, diesen Speicherbereich zu lesen. So kann z. B. in Programmen festgestellt werden, ob der benutzte Computer ein CPC 464 oder der neue CPC 664 ist. Programme, die sehr tief auf Maschinenebene hinabsteigen, können damit schon beim Programmstart feststellen, ob sie auf dem CPC 664 aufgrund des geänderten ROMs überhaupt lauffähig sind, und notfalls eine Fehlermeldung ausgeben.

```
10a%=0:b%=0:c%=0
20 IROMPEEK,49153,@a%:IROMPEEK,49154,@b%:
IROMPEEK,49155,@c%
30 IF a%=1 and b%=0 and c%=0 THEN PRINT
"Ein CPC 464 also!":END
40 PRINT "Sie benutzen einen CPC 664"
```

Hier müssen Sie noch einiges beachten: Die Übergabe von Daten an Maschinenprogramme geschieht mühelos, allerdings ist der umgekehrte Weg nicht ganz so einfach. Man kann mit dem Klammeraffen (ASCII-Code 64) statt einer Variablen deren Adresse im Basic-RAM übermitteln, die von Maschinenprogrammen übernommen wird. Durch direkte Veränderung der Werte an der Zieladresse können auch aus Maschinenprogrammen heraus Basic-Variablen neue Werte zugewiesen werden. Daszu muß allerdings die Variable bereits definiert sein, z. B. **a%=123,** und es darf sich in der vorliegenden Version des Maschinencodes nur um Integervariablen handeln. Die Integerdefinition kann entweder mit DEFINT geschehen oder durch ein Prozentzeichen nach dem Variablennamen.

IROMDEEK,Adresse, Variable
Dieser Befehl ist eine Mischung des DEEK- und des ROMPEEK-Befehls, denn hiermit können 16-Bit-Werte aus dem ROM gelesen werden. Somit läßt sich das Identifikationsprogramm, das als Anwendungsbeispiel des ROMPEEK-Befehls fungiert, verkürzen:

```
10 a%=0:b%=0
20 IROMDEEK,49152,@a%:IROMDEEK,49154,@b%
30 IF a%=&8001 AND b%=0 THEN PRINT
   "Ein CPC 464":END
40 PRINT "Ein CPC 664":END
```


IXCALL,Adresse,ROM-Status,Akku,BC,DE,HL
Der normale CALL-Befehl zum Aufruf von Maschinenroutinen hat zwei Nachteile: Erstens können nur Routinen angesprungen werden, die im RAM stehen; und zweitens ist es nicht möglich, beim Aufruf bestimmte Werte für die Register zu übergeben.
XCALL steht für **Extended CALL** und ermöglicht es, Programme irgendwo im ROM oder RAM, ja sogar im Floppy-ROM oder einem der 252 anschließbaren Erweiterungs-ROMs anzusprechen. Die Speicherkonfiguration wird durch das ROM-Status-Byte im XCALL-Befehl selektiert:

ROM-Status	&0000-&3FFF	&C000-&FFFF
252	Betriebssystem-ROM	Basic-ROM
253	RAM-Bereich	Basic-ROM
254	Betriebssystem-ROM	RAM-Bereich
255	RAM-Bereich	RAM-Bereich

Die übrigen Werte von 0 bis 251 erlauben die Adressierung der Zusatz-ROMs, die auf den Expansion Port aufgesteckt werden können. Bisher sind hier z. B. der Floppy-Controller sowie diverse Assembler als ROM-Packs erhältlich.
Mit dem XCALL-Befehl können aber auch vorgegebene Werte für die Hauptregister übergeben werden, nämlich der Akku und die Doppelregister BC, DE und HL. Somit haben Sie die Möglichkeit, z. B. in auszutestende Maschinenprogramme an mehren Stellen mit vorgegebenen Registerwerten einzuspringen. Eine weitere Einsatzmöglichkeit sind nützliche ROM-Routinen – wie z. B. folgende:
- IXCALL,&BB45,252,0,0,0,0 löst beim Drücken der ESCAPE-Taste einen Reset des Computers aus, was wohl hauptsächlich als Programmschutz Verwendung finden wird. Der Befehl ist aber nur in laufenden Programmen wirksam, da die Abfrage der ESC-Taste im Direktmodus anders funktioniert.
- IXCALL,&BC05,252,0,0,0,Diff ermöglicht es, die Basisadresse des Bildschirms in Abständen von zwei Bytes zu verschieben. Bei geschickter Anwendung kann damit ein Rechts/Links-Scrolling erreicht werden:
FOR i=0 TO 2048 STEP 2:CALL &BD19:IXCALL,&BC05,
252,0,0,0,i:NEXT i
- IXCALL,&BC08,252,Basis,0,0,0 erlaubt die Verschiebung des Bildschirmspeichers in 16KB-Blöcken; mögliche Werte sind also &00 (Startadresse &0000), &40 (Startadresse & 4000), &80 (Startadresse &8000) oder der Standardwert &C0 für den Beginn des Bildschirmspeichers an der Adresse &C000. Zur Verwendung in Basic sind aber nur die Werte &C0 und &40 zulässig, da in allen anderen Bereichen wichtige Informationen für das Betriebssystem stehen. Beschränken Sie sich also auf einen 16K-Speicher für Ihre Basic-Programme, so können maximal zwei vollständige Grafikbildschirme gleichzeitig im Speicher untergebracht werden:

```
10 MEMORY &3FFF ' Speicher für Grafik reservieren
20 MODE 1:FOR i=1 TO 25:PRINT STRING$(40,42):NEXT i
30 IXCALL,&BC08,252,&40,0,0,0
40 FOR i=0 TO 640 STEP 4:MOVE i,0:DRAW i,640:NEXT i
50 ' Bildschirme dauernd austauschen
60 FOR i=1 TO 1000:IXCALL,&BC08,252,&40,0,0,0
70 IXCALL,&BC08,252,&C0,0,0,0:NEXT i:END
```

- IXCALL,&BD10,252,0,0,Time1,Time2: Hiermit kann der Zeitgeber, der mit der BASIC-Funktion TIME zwar gelesen, aber nicht verändert werden kann, auch auf eine bestimmte Uhrzeit eingestellt werden – Anwendungsfall: Simulation einer Echtzeituhr. Dieser Timer ist ein 4-Byte-Zähler; dies bedeutet, daß zur Datenübergabe zwei Doppelregister benötigt werden, nämlich DE und HL. Mit dem folgenden Betriebssystem-Aufruf kann der Timer zurückgesetzt werden:
IXCALL,&BD10,252,0,0,0,0:PRINT TIME
- IXCALL,&B91B,255,0,BC,DE,HL verschiebt analog zum Z80-Maschinenbefehl LDIR ganze Speicherblöcke bis zu 64 KByte im Speicher – aber immer nur im RAM. Wie beim LDIR-Befehl müssen die 16-Bit-Register BC, DE und HL mit der Blocklänge, der Quell- und der Zieladresse versorgt werden. Hiermit ist eine andere Möglichkeit gegeben, Grafikbildschirme intern abzuspeichern:

```
10 MODE 1:MEMORY 22999:PRINT "Bildschirmdarstellung"
   ,,,,STRING$(40,255)
20 RANDOMIZE 0
30 MOVE 0,0:FOR i=0 TO 100:DRAW RND*640,RND*300,
   RND*3:NEXT i
40 FOR i=0 TO 2000:NEXT i
50 IXCALL,&B91B,255,0,16383,23000,49152
60 CLS:FOR i=0 TO 2000:NEXT i
70 IXCALL,&B91B,255,0,16383,49152,23000
80 LOCATE 1,10
```

- IXCALL,&B91E,255,0,BC,DE,HL entspricht dem vorhergehenden LDIR-Aufruf, allerdings wird hier der LDIR-Befehl ausgeführt – ebenfalls mit ausgeblendeten ROMs.
- IXCALL,&CA94,253,0,0, Code,0 zeigt, daß auch Routinen im Basic-ROM aufrufbar sind. Dieser Befehl gestattet beispielsweise wie der ERROR-Befehl die Ausgabe beliebiger Fehlermeldungen:

IXCALL,&CA94,253,0,0,10,0 gibt einen Fehler
"ARRAY ALREADY DIMENSIONED" aus.

Martin Kotulla

```
100 '  ********************************
110 ' **********************************
120 ' *                                *
130 ' *   BASIC-LOADER: Machine Kit    *
140 ' *                                *
150 ' **********************************
160 '  ********************************
170 '
180 '   (C) Martin Kotulla  18.6.1985
190 '   ----------------------------
200 '
```

```
210 MEMORY 41779 ' Speicherobergrenze festlegen
220 FOR i=41780 TO 41994:READ a:sum=sum+a:NEXT i
230 IF sum=24605 THEN 250
240 PRINT CHR$(7);"* Fehler in den DATAs":PRINT:LIST 270-:END
250 RESTORE:FOR i=41780 TO 41994:READ a:POKE i,a:NEXT i
260 CALL 41783 ' RSX-Erweiterung einbinden
270 DATA &52,&4F,&4D,&21,&7E,&A3,&01,&40,&A3,&C3,&D1,&BC,&54,&A3,&C3,&82
280 DATA &A3,&C3,&93,&A3,&C3,&9D,&A3,&C3,&AD,&A3,&C3,&C6,&A3,&C3,&E3,&A3
290 DATA &56,&49,&44,&45,&4F,&2E,&52,&45,&C7,&53,&4F,&55,&4E,&44,&2E,&52
300 DATA &45,&C7,&44,&4F,&4B,&C5,&52,&4F,&4D,&50,&45,&45,&CB,&52,&4F,&4D
310 DATA &44,&45,&45,&CB,&58,&43,&41,&4C,&CC,&00,&20,&20,&00,&00,&DD,&7E
320 DATA &02,&01,&00,&BC,&ED,&79,&DD,&7E,&00,&01,&00,&BD,&ED,&79,&C9,&DD
330 DATA &7E,&02,&DD,&4E,&00,&CD,&34,&BD,&C9,&DD,&66,&03,&DD,&6E,&02,&DD
340 DATA &7E,&00,&77,&23,&DD,&7E,&01,&77,&C9,&DD,&66,&03,&DD,&6E,&02,&DD
350 DATA &56,&01,&DD,&5E,&00,&CD,&00,&B9,&CD,&06,&B9,&7E,&EB,&77,&23,&36
360 DATA &00,&C9,&DD,&66,&03,&DD,&6E,&02,&DD,&56,&01,&DD,&5E,&00,&CD,&00
370 DATA &B9,&CD,&06,&B9,&7E,&EB,&23,&77,&EB,&23,&7E,&EB,&2B,&77,&C9,&DD
380 DATA &6E,&0A,&DD,&66,&0B,&22,&34,&A3,&DD,&7E,&08,&32,&36,&A3,&DD,&6E
390 DATA &00,&DD,&66,&01,&DD,&5E,&02,&DD,&56,&03,&DD,&4E,&04,&DD,&46,&05
400 DATA &DD,&7E,&06,&DF,&34,&A3,&C9
```

```
1000 ';  ************************************************************************
1010 '; ************************************************************************
1020 '; *                                                                      *
1030 '; *   Machine Kit: RSX-Erweiterung zur maschinennahen Programmierung     *
1040 '; *                                                                      *
1050 '; ************************************************************************
1060 ';  ********************************************************************
1070 ';
1080 '; -------------------- (C) Martin Kotulla 18.6.1985 --------------------
1090 ';
1100 '; Neue Befehle: "|VIDEO.REG,Register,Wert"
1110 ';               "|SOUND.REG,Register,Wert"
1120 ';               "|DOKE,Adresse,Wert"
1130 ';               "|ROMPEEK,Adresse,@Variable"
1140 ';               "|ROMDEEK,Adresse,@Variable"
1150 ';               "|XCALL,Adresse,ROMstatus,Akku,BC,DE,HL"
1160 ';
1170 '        ORG  41780            ; Oder wohin man will ....
1180 '
1190 ' UROMON EQU  &B900            ; KL UPPER ROM ENABLE: Basic-ROM einschalten
1200 ' LROMON EQU  &B906            ; KL LOWER ROM ENABLE: OS-ROM einschalten
1210 ' SNDREG EQU  &BD34            ; MC Sound Register
1220 ' LOGEXT EQU  &BCD1            ; RSX-Modul einbinden
1230 '
1240 ' XCALLV DEFS 2                ; Sprungvektor fuer XCALL
1250 ' CONFIG DEFS 1                ; RAM/ROM-Konfigurationsbyte fuer XCALL
1260 '
1270 '; ***** RSX-ERWEITERUNG EINBINDEN ***************************************
1280 '
```



```
1290 ' INIT   LD   HL,KERNEL       ; Zeiger auf Puffer zur Kernel-Verwaltung
1300 '        LD   BC,RSX          ; Zeiger auf Sprungtabelle fuer RSX-Module
1310 '        JP   LOGEXT          ; RSX-Module einbinden, Basic-Ruecksprung
1330 '
1340 ' RSX    DEFW RSXTAB          ; Zeiger auf Namenstabelle der RSXen
1350 '        JP   VWRITE          ; Sprung in RSX: VIDEO.WRITE
1360 '        JP   SWRITE          ; Sprung in RSX: SOUND.WRITE
1370 '        JP   DOKE            ; Sprung in RSX: DOKE
1380 '        JP   ROMPK           ; Sprung in RSX: ROMPEEK
1390 '        JP   ROMDK           ; Sprung in RSX: ROMDEEK
1400 '        JP   XCALL           ; Sprung in RSX: XCALL
1410 '
1420 ' RSXTAB DEFM "VIDEO.RE"
1430 '        DEFB &C7
1440 '        DEFM "SOUND.RE"
1450 '        DEFB &C7
1460 '        DEFM "DOK"
1470 '        DEFB &C5
1480 '        DEFM "ROMPEE"
1490 '        DEFB &CB
1500 '        DEFM "ROMDEE"
1510 '        DEFB &CB
1520 '        DEFM "XCAL"
1530 '        DEFB &CC
1540 '        DEFB &00             ; Ende der Tabelle
1550 '
1560 ' KERNEL DEFS &04              ; Hilfsspeicher fuer Kernel-ROM
1570 '
1580 '; ****** VIDEO.WRITE ********************************************************
1590 '
1600 ' VWRITE LD   A,(IX+2)        ; Lowbyte der Registernummer holen
1610 '        LD   BC,&BC00        ; Adressport des Video-Chips
1620 '        OUT  (C),A           ; Registernummer auf Adressport geben
1630 '        LD   A,(IX+0)        ; Lowbyte des Schreibwerts holen
1640 '        LD   BC,&BD00        ; Datenport des Video-Chips
1650 '        OUT  (C),A           ; Schreibwert auf Datenport geben
1660 '        RET                  ; Ruecksprung nach Basic
1670 '
1680 '; ***** SOUND.WRITE ********************************************************
1690 '
1700 ' SWRITE LD   A,(IX+2)        ; Registernummer holen
1710 '        LD   C,(IX+0)        ; Daten fuer Sound-Chip holen
1720 '        CALL SNDREG          ; MC Sound Register
1730 '        RET
1740 '
1750 '; ***** DOKE ***************************************************************
1760 '
1770 ' DOKE   LD   H,(IX+3)        ; Highbyte der zu schreibenden Adresse holen
1780 '        LD   L,(IX+2)        ; Lowbyte der zu schreibenden Adresse holen
1790 '        LD   A,(IX+0)        ; Lowbyte des Schreibwerts holen
```

```
1800 '         LD   (HL),A          ; An Adresse schreiben
1810 '         INC  HL              ; Adresse um eins erhoehen
1820 '         LD   A,(IX+1)        ; Highbyte des Schreibwerts holen
1830 '         LD   (HL),A          ; Highbyte an Adresse schreiben
1840 '         RET                  ; Ruecksprung nach Basic
1850 '
1860 '; ***** ROMPEEK ***************************************************
1870 '
1880 ' ROMPK   LD   H,(IX+3)        ; Highbyte der zu lesenden Adresse holen
1890 '         LD   L,(IX+2)        ; Und das Lowbyte holen
1900 '         LD   D,(IX+1)        ; Highbyte der Variablenadresse holen
1910 '         LD   E,(IX+0)        ; Lowbyte ebenso holen
1920 '         CALL UROMON          ; Basic-ROM einschalten
1930 '         CALL LROMON          ; Betriebssystem-ROM einschalten
1940 '         LD   A,(HL)          ; Ein Byte an der Adresse lesen
1950 '         EX   DE,HL           ; HL:=Variablenadresse
1960 '         LD   (HL),A          ; Den gelesenen Wert ins Variablen-Lowbyte
1970 '         INC  HL              ; Zeiger auf zweites Byte des Variablenwerts
1980 '         LD   (HL),0          ; Dorthin ins Highbyte Null schreiben
1990 '         RET                  ; Ruecksprung nach Basic
2000 '
2010 '; ***** ROMDEEK ***************************************************
2020 '
2030 ' ROMDK   LD   H,(IX+3)        ; Highbyte der zu lesenden Adresse holen
2040 '         LD   L,(IX+2)        ; Ebenso das Lowbyte holen
2050 '         LD   D,(IX+1)        ; Highbyte der Variablenadresse holen
2060 '         LD   E,(IX+0)        ; Ebenso das Lowbyte holen
2070 '         CALL UROMON          ; Basic-ROM einschalten
2080 '         CALL LROMON          ; Betriebssystem-ROM einschalten
2090 '         LD   A,(HL)          ; Ein Byte an der Adresse lesen
2100 '         EX   DE,HL           ; HL:=Variablenadresse
2110 '         INC  HL              ; Zuerst Zeiger auf das Variablen-Highbyte
2120 '         LD   (HL),A          ; Den gelesenen Wert ins Highbyte
2130 '         EX   DE,HL           ; HL:=Zu lesende Adresse
2140 '         INC  HL              ; Zeiger auf zweites Byte des 16 Bit-Worts
2150 '         LD   A,(HL)          ; Dortigen 8 Bit-Wert lesen
2160 '         EX   DE,HL           ; HL:=Zeiger auf Variable (2.Byte)
2170 '         DEC  HL              ; Zeiger auf erstes Byte des Variablenwerts
2180 '         LD   (HL),A          ; Dorthin den gelesenen Wert schreiben
2190 '         RET                  ; Ruecksprung nach Basic
2200 '
2210 '; ***** XCALL *****************************************************
2220 '
2230 ' XCALL   LD   L,(IX+10)       ; Adresse der Routine holen: Lowbyte
2240 '         LD   H,(IX+11)       ; Ebenso das Highbyte
2250 '         LD   (XCALLV),HL     ; Sprungadresse in den XCALLV-Vektor
2260 '         LD   A,(IX+8)        ; RAM/ROM-Statusbyte holen
2270 '         LD   (CONFIG),A      ; Fuer den RST3-Befehl Speicherkonfiguration
2280 '
2290 '         LD   L,(IX+0)        ; \  Wert fuer das HL-Register
```



```
2300 '         LD   H,(IX+1)        ; /  aus dem XCALL-Befehl holen
2310 '
2320 '         LD   E,(IX+2)        ; \  Wert fuer das DE-Register
2330 '         LD   D,(IX+3)        ; /  aus dem XCALL-Befehl holen
2340 '
2350 '         LD   C,(IX+4)        ; \  Wert fuer das BC-Register
2360 '         LD   B,(IX+5)        ; /  aus dem XCALL-Befehl holen
2370 '
2380 '         LD   A,(IX+6)        ; Akkuwert aus dem XCALL-Befehl holen
2390 '         RST  &0018
2400 '         DEFW XCALLV          ; Pointer auf XCALL-Vektor
2410 '         RET                  ; Ruecksprung nach Basic
2420 '
2430 '         END                  ; ***************************************
```

Tabelle 1: Die Register des Video-Chips MOTOROLA 6845 CRTC

Register		Kurzbezeichnung	Normwert	Funktion
&00	0	Horizontal Total	63	Zahl der Zeichen pro vollständige Zeile; der Wert ist etwa 1.5 mal so groß wie die tatsächliche Zahl der Zeichen pro Zeile.
&01	1	Horiz Displayed	40	Zahl der tatsächlich angezeigten Zeichen pro Bildschirmzeile.
&02	2	Horiz Sync Position	46	Zeitpunkt des horizontalen Sync-Impulses. Durch Veränderung des Wertes Links/Rechts-Scrolling möglich.
&03	3	VSync/HSync Width	140	Breite des Vert-Sync- und Horiz-Sync-Impulses.
&04	4	Vertical Total	38	Zahl der Zeilen auf dem Monitor; vgl. Erläuterungen zu R0 und R1.
&05	5	Vert Total Adjust	0	Genauer Abgleich der Bildposition auf dem Monitor; durch Veränderung des Registerwerts ist ein Softscrolling nach oben oder unten möglich.
&06	6	Vertical Displayed	25	Zahl der tatsächlich angezeigten Zeilen auf dem Bildschirm. Anwendungsbeispiel im Text.
&07	7	Vert Sync Position	30	Zeitpunkt des vertikalen Sync-Impulses. Durch Veränderung des Wertes ist eine Bildverschiebung nach unten oder oben möglich.
&08	8	Interlace & Skew	0	Die Bits 0 und 1 bestimmen, ob die Bildschirmdarstellung mit Interlace und Skew erfolgen soll.
&09	9	Max Raster Address	7	Anzahl der Rasterzeilen der Zeichen auf dem Bildschirm.
&0A	10	Cursor Start Raster	–	Rasterzeile des Cursors und dessen Darstellungsmodus (blinkend, fest, Blinkgeschwindigkeit)
&0B	11	Cursor End Raster	–	Rasterzeile, auf der der Cursor endet.
&0C	12	Start Address High	–	Beginn des Bildschirmspeichers (High).
&0D	13	Start Address Low	–	Beginn des Bildschirmspeichers (Low).
&0E	14	Cursor High	–	High-Byte der Cursorposition.
&0F	15	Cursor Low	–	Low-Byte der Cursorposition.

Es gibt noch zwei weitere Register, die derzeit von der Schneider-Software nicht verwendet werden: POS STROBE und LIGHT PEN.

Tabelle 2: Die Register des Sound-Chips GENERAL INSTRUMENTS AY-3-8912

Register	Kurzbezeichnung	Funktion
R00	Period A Low	Feinabstimmung der Periodendauer für Kanal A

R01	Period A High	Grobabstimmung der Periodendauer für Kanal A
R02	Period B Low	Feinabstimmung der Periodendauer für Kanal B
R03	Period B High	Grobabstimmung der Periodendauer für Kanal B
R04	Period C Low	Feinabstimmung der Periodendauer für Kanal C
R05	Period C High	Grobabstimmung der Periodendauer für Kanal C
R06	Noise Period	Periodendauer des Geräuschgenerators
R07	Multi Fctn	Mehere Funktionen: - Kanal A, B und C einschalten - Rauschen zu Kanal A, B und C - Port A als Ein- oder Ausgang (Tastatur)
R08	Volume A	Lautstärke von Kanal A
R09	Volume B	Lautstärke von Kanal B
R10	Volume C	Lautstärke von Kanal C
R11	Env Period Low	Hüllkurven-Periodendauer (Lowbyte)
R12	Env Period High	Hüllkurven-Periodendauer (Highbyte)
R13	Env Shape	Hüllkurvenform

„BASIC +"-Programm »Für den CPC 464«

BASIC + ist ein RSX-Utility, welches sich gleichermaßen für den Schneider CPC 464 als auch für den neuen CPC 664 eignet. RSX bedeutet Resistant System Extention. Das bedeutet, daß neue Befehle erzeugt werden, die bis zum Ausschalten oder RESET des Computers zur Verfügung stehen. Doch dazu später mehr. Das vorliegende Programm erzeugt folgende Befehle: :SCROLLUP – SCROLLt den Bildschirm eine Zeile nach oben. :SCROLLDOWN–wie SCROLLUP, jedoch nach unten. :BREAKOFF – Programme lassen sich nach Anwendung dieses Befehls in keiner Weise mehr BREAKen. Die lästige Befehlsfolge ON BREAK GOSUB-RETURN entfällt also. ACHTUNG! Vor Probelauf das Programm ohne Breakschutz abspeichern. :RESETOFF – Der Computer läßt sich nicht mehr durch RESET zurücksetzen. :RESETON – RESET-Funktion des Computers wird wieder eingeschaltet. :GETCHAR – Dies ist der aufwendigste Befehl. Er dient dazu Zeichen an einer angegebenen Position vom Bildschirm zu lesen. Hierzu sind aber noch die Parameter X,Y,V anzugeben, so daß der Befehl mit :GETCHAR,-X,Y,V angesprochen wird.
X und Y geben die Bildschirmposition an, wobei aber auf den MODE geachtet werden muß. Bei V wird es schon komplizierter. V bezeichnet die Speicherstelle einer Integervariablen. Diesen erhält man, indem man vor die Variable den 'Klammeraffen' setzt. (Der Klammeraffe soll im Folgenden durch '#' ersetzt werden.) Die Variable muß vorher aber auch noch definiert werden, da sonst ja noch kein Speicherplatz existiert. Man schreibt also zum Beispiel i%=0, danach :GETCHAR,-X,Y,#i%. Danach steht der ASCII-Wert des Zeichens in i%. Das '%' steht deshalb, weil die Variable vom Typ Integer sein muß. :BAUD,BR – Dieser Befehl ersetzt den SPEED WRITE Befehl des Schneider CPC. Sie haben nun die Möglichkeit, Programme nicht nur mit 1000 oder 2000 Baud, sondern mit einer Baudrate zwischen ca. 750 und 3600 abzuspeichern. Andere Baudraten sind nicht möglich, da es der CPC sonst nicht schafft, das Signal zu synchronisieren. Doch ausprobieren. :WAIT – Der Programmablauf wird solange unterbrochen, bis eine Taste gedrückt wird. :RECON – Schaltet Datenrecorder ein. :RECOFF – Schaltet Datenrecorder aus. Achtung! Der Doppelpunkt vor den Befehlen ist das Zeichen über dem Klammeraffen.

Zur Eingabe des Programms:
Besitzer eines Assemblers können direkt die Mnemoniks eingeben. Alle anderen sollten mit Hilfe des Eingabeprogramms den hexadezimalen Maschinencode (in der zweiten Spalte des Assemblerprogramms) eingeben. Ist dies geschehen, gibt man das Ladeprogramm ein und speichert alles mit SAVE"BASIC+":SAVE"BASIC+MC",B,&A100,&E0,&A100 ab. Nach Laden des Programms mit RUN" stehen nun die Befehle zur Verfügung. Abschließend noch etwas zur Erzeugung von eigenen RSX-Befehlen, am Beispiel des vorliegenden Programms. Die RSX Routine wird mit CALL #BCD1 angesprochen. Ihr übergeben werden in BS die Adresse der Sprungtabelle (Z.1320) u. in HL die eines vier Bytes großen Buffers. Die ersten zwei Bytes der Sprungtabelle bezeichnen den Anfang der Befehlstabelle (Z.1370). In dieser (Z.1490) werden die Befehlsnamen entsprechend der Reihenfolge der Sprungtabelle aufgelistet. Das Ende eines Befehls wird dadurch signalisiert, daß im letzten Buchstaben Bit 7 gesetzt ist (+#80). Das Ende der Befehlstabelle wird durch ein 0 signalisiert (Z.1700).

```
100 REM
***** BASIC + Ladeprogramm *****
110 MODE 0
120 LOCATE 6,11:PRINT"***********"
130 LOCATE 6,12:PRINT"* BASIC + *"
140 LOCATE 6,13:PRINT"***********"
150 LOCATE 1,20:PEN 8:PRINT"(C) '85 by K.Micheli"
160 PRINT:PEN 2:PRINT"is loading ...."
170 MEMORY &A0FF
180 LOAD"!"
190 CALL &A100
200 MODE 1:PEN 1:PRINT"BASIC + READY":NEW
```



```
100 REM ***** MC-Eingabeprogramm zu BASIC + *****
110 adr=&A100
120 MEMORY adr-1
130 PRINT HEX$(adr);"---->";:INPUT a$
140 IF LEN(a$)/2<>INT(LEN(a$)/2) THEN PRINT"ERROR":GOTO 130
150 FOR i=1 TO LEN(a$) STEP 2
160   POKE adr,VAL("&"+MID$(a$,i,2))
170   adr=adr+1
180   IF adr=&A1DA THEN PRINT"ENDE":END
190 NEXT i
200 GOTO 130
```

Hexadezimaler Maschinencode

```
01 0A A1 21 DA A1 CD D1 BC C9 2A A1 C3 6E A1 C3 77 A1 C3 80
A1 C3 84 A1 C3 A8 A1 C3 C2 A1 C3 C6 A1 C3 CA A1 C3 CE A1 C3
D4 A1 53 43 52 4F 4C 4C 55 D0 53 43 52 4F 4C 4C 44 4F 57 CE
42 52 45 41 4B 4F 46 C6 47 45 54 43 48 41 D2 42 41 55 C4 57
41 49 D4 52 45 43 4F CE 52 45 43 4F 46 C6 52 45 53 45 54 4F
46 C6 52 45 53 45 54 4F CE 00 06 01 3A 90 B2 CD 4D BC C9 06
00 3A 90 B2 CD 4D BC C9 CD 48 BB C9 DD 5E 00 DD 56 01 DD 6E
02 DD 66 04 D5 E5 CD 78 BB 22 DE A1 E1 CD 75 BB CD 60 BB D1
12 2A DE A1 CD 75 BB C9 DD 6E 00 DD 66 01 11 0B 00 CD C1 BD
E5 D1 21 5F 76 CD C1 BD 3E 0A CD 68 BC C9 CD 06 BB C9 CD 6E
BC C9 CD 71 BC C9 3E C9 32 EE BD C9 3E C3 32 EE BD C9
```

```
                1000 ************************************************
                1010 ***              BASIC + (V 2.1)             ***
                1020 ***   (C) 1985 by Klaus Micheli, Troisdorf   ***
                1030 ************************************************
                1040 *** Syntax der Befehle:!SCROLLUP             ***
                1050 ***                    !SCROLLDOWN           ***
                1060 ***                    !BREAKOFF             ***
                1070 ***                    !RESETOFF             ***
                1080 ***                    !RESETON              ***
                1090 ***                    !GETCHAR,X,Y,V        ***
                1100 ***                    !BAUD,BR              ***
                1110 ***                    !WAIT                 ***
                1120 ***                    !RECON                ***
                1130 ***                    !RECOFF               ***
                1140 ************************************************
                1150
                1155
BB06            1160 WAIT:    EQU  #BB06
BB48            1170 BRKOFF:  EQU  #BB48
BB78            1180 GETCRS:  EQU  #BB78
BB75            1190 SETCRS:  EQU  #BB75
BB60            1200 RDCHAR:  EQU  #BB60
BC4D            1210 SCRHW:   EQU  #BC4D
BC68            1220 BAUD:    EQU  #BC68
BCD1            1230 RSX:     EQU  #BCD1
B290            1240 PAPER:   EQU  #B290
BDC1            1250 DIV:     EQU  #BDC1
BDEE            1260 RESET:   EQU  #BDEE
BC6E            1270 RON:     EQU  #BC6E
BC71            1280 ROFF:    EQU  #BC71
                1290
A100            1300          ORG  #A100
                1310
```

```
A100  010AA1    1320          LD   BC,COMTAB          ;Sprungtabelle und
A103  21DAA1    1330          LD   HL,BUF             ;Buffer an
A106  CDD1BC    1340          CALL RSX                ;RSX-Routine uebergeben
A109  C9        1350          RET
                1360
A10A  2AA1      1370 COMTAB:  DEFW BEFTAB             ;Zeiger auf
A10C  C36EA1    1380          JP   SCRU               ;Befehlstabelle
A10F  C377A1    1390          JP   SCRD
A112  C380A1    1400          JP   BREAKA
A115  C384A1    1410          JP   GETCHR
A118  C3A8A1    1420          JP   SPEED
A11B  C3C2A1    1430          JP   WARTE
A11E  C3C6A1    1440          JP   RECON
A121  C3CAA1    1450          JP   RECOFF
A124  C3CEA1    1460          JP   REOFF
A127  C3D4A1    1470          JP   REON
                1480
A12A  5343524F  1490 BEFTAB:  DEFM "SCRO"            ;Definition der Befehle
A12E  4C4C55    1495          DEFM "LLU"
A131  D0        1500          DEFB "P"+#80
A132  5343524F  1510          DEFM "SCRO"
A136  4C4C444F  1515          DEFM "LLDO"
A13A  57        1517          DEFM "W"                ;.
Hisoft GENA3.1 Assembler. Page     2.

A13B  CE        1520          DEFB "N"+#80
A13C  42524541  1530          DEFM "BREA"
A140  4B4F46    1535          DEFM "KOF"
A143  C6        1540          DEFB "F"+#80
A144  47455443  1550          DEFM "GETC"
A148  4841      1555          DEFM "HA"
A14A  D2        1560          DEFB "R"+#80
A14B  424155    1580          DEFM "BAU"
A14E  C4        1590          DEFB "D"+#80
A14F  574149    1600          DEFM "WAI"
A152  D4        1610          DEFB "T"+#80
A153  5245434F  1620          DEFM "RECO"
A157  CE        1630          DEFB "N"+#80
A158  5245434F  1640          DEFM "RECO"
A15C  46        1645          DEFM "F"
A15D  C6        1650          DEFB "F"+#80
A15E  52455345  1660          DEFM "RESE"
A162  544F46    1665          DEFM "TOF"
A165  C6        1670          DEFB "F"+#80
A166  52455345  1680          DEFM "RESE"
A16A  544F      1685          DEFM "TO"
A16C  CE        1690          DEFB "N"+#80
A16D  00        1700          DEFB #00                ;Ende von Beftab.
                1710
A16E  0601      1720 SCRU:    LD   B,#01              ;Befehl SCROLLUP
A170  3A90B2    1730          LD   A,(PAPER)
A173  CD4DBC    1740          CALL SCRHW
A176  C9        1750          RET
                1760
A177  0600      1770 SCRD:    LD   B,#00              ;Befehl SCROLLDOWN
A179  3A90B2    1780          LD   A,(PAPER)
A17C  CD4DBC    1790          CALL SCRHW
A17F  C9        1800          RET
                1810
```



```
A180  CD48BB    1820 BREAKA:  CALL BRKOFF              ;Befehl BREAKOFF
A183  C9        1830          RET
                1840
A184  DD5E00    1850 GETCHR:  LD   E,(IX+#00)          ;Befehl GETCHAR,X,Y,V
A187  DD5601    1860          LD   D,(IX+#01)
A18A  DD6E02    1870          LD   L,(IX+#02)
A18D  DD6604    1880          LD   H,(IX+#04)
A190  D5        1890          PUSH DE
A191  E5        1900          PUSH HL
A192  CD78BB    1910          CALL GETCRS
A195  22DEA1    1920          LD   (OLDPOS),HL
A198  E1        1930          POP  HL
A199  CD75BB    1940          CALL SETCRS
A19C  CD60BB    1950          CALL RDCHAR
A19F  D1        1960          POP  DE
A1A0  12        1970          LD   (DE),A
A1A1  2ADEA1    1980          LD   HL,(OLDPOS)
A1A4  CD75BB    1990          CALL SETCRS
A1A7  C9        2000          RET
                2010
A1A8  DD6E00    2020 SPEED:   LD   L,(IX+#00)          ;Befehl BAUD,BR
A1AB  DD6601    2030          LD   H,(IX+#01)
A1AE  110B00    2040          LD   DE,#000B
A1B1  CDC1BD    2050          CALL DIV                ;.
Hisoft GENA3.1 Assembler. Page     3.

A1B4  E5        2060          PUSH HL
A1B5  D1        2070          POP  DE
A1B6  215F76    2080          LD   HL,30303
A1B9  CDC1BD    2090          CALL DIV
A1BC  3E0A      2100          LD   A,#0A
A1BE  CD68BC    2110          CALL BAUD
A1C1  C9        2120          RET
                2130
A1C2  CD06BB    2140 WARTE:   CALL WAIT               ;Befehl WAIT
A1C5  C9        2150          RET
                2160
A1C6  CD6EBC    2170 RECON:   CALL RON                ;Befehl RECON
A1C9  C9        2180          RET
                2190
A1CA  CD71BC    2200 RECOFF:  CALL ROFF               ;Befehl RECOFF
A1CD  C9        2210          RET
                2220
A1CE  3EC9      2230 REOFF:   LD   A,#C9              ;Befehl RESETOFF
A1D0  32EEBD    2240          LD   (RESET),A
A1D3  C9        2250          RET
                2260
A1D4  3EC3      2270 REON:    LD   A,#C3              ;Befehl RESETON
A1D6  32EEBD    2280          LD   (RESET),A
A1D9  C9        2290          RET
                2300
A1DA            2310 BUF:     DEFS #04                ;Buffer fuer RSX-Routine
A1DE            2320 OLDPOS:  DEFS #02                ;Speicher fuer Cursorposi
tion

Pass 2 errors: 00

Table used:   333  from   400
```

Apple-Programm:
„Digital Sound Editor (DSE)

Unter dem Begriff DSE kann sich wohl keiner was bestimmtes vorstellen, deshalb wird erst einmal erklärt, was das Programm überhaupt kann:

1. DSE macht es möglich, dem Lautsprecher des APPLE jedes beliebige Geräusch zu entlocken (Explosionen, Weltraumsounds, ...). Dies funktioniert ohne Zusatzhardware!
2. Das Geräusch kann in der hochauflösenden Grafik mit einem komfortablen Editor editiert und verändert werden.
3. Es können auch Geräusche von außen digitalisiert und editiert werden.
4. Mathematische Funktionen können mit dem Function Sound Creater (FSC) in Töne umgewandelt werden.
5. Zwei Geräusche können gleichzeitig editiert und auch gemischt werden.
6. Der Einbau der Geräusche in eigene Spiel- oder sonstige Programme bereitet keinerlei Schwierigkeiten.

Dies sind doch recht erstaunliche Dinge für einen APPLE II Computer – oder?
Nun zu den Funktionen im einzelnen:

1. LADEN
Diese Funktion lädt einen vorher auf Diskette abgespeicherten Sound in den Arbeitsspeicher des APPLE. Das Geräusch wird in die Seite geladen, die vorher editiert wurde, sonst in Seite 1.

2. SAVEN
Hiermit kann man ein selbst kreiertes Geräusch auf Diskette abspeichern. Es wird wie bei Funktion 1 die aktuelle Soundseite gesaved.

3. Editieren
Nun kommen wir zum Supereditor des DSE-Programms. Nach Anwählen des Menüpunktes 3 wird eine HGR-Seite gezeigt und der momentane (Sound-)speicherinhalt dargestellt. Rechts ist eine Leiste, auf der verschiedene Editierzustände angezeigt werden. Ein Pfeil zeigt den Strich (bzw. die Speicherstelle), der gerade editiert werden könnte. Dieser Pfeil wird mit dem Joystick bewegt, bei Druck auf die Taste 0 des Joysticks kann man die einzelnen Striche verlängern oder verkürzen, wodurch das Geräusch verändert wird. Ein Druck auf die Taste 'P' (PLAY) spielt den Sound vor. Die Abspielgeschwindigkeit kann mit den Tasten '<--' und '-->' vergrößert oder verkleinert werden. Die momentane Geschwindigkeit steht im Schriftfeld rechts oben. Der Geschwindigkeitsbereich geht von 0-255. Um schneller eine andere Geschwindigkeit zu erreichen, kann die Taste '2' gedrückt werden. Ein '$' wird im Schriftfeld angezeigt, und die Geschwindigkeit wechselt dann in 10er Schritten. Die Taste '1' macht dies wieder rückgängig.
Die Taste '+' bewirkt, daß nun die zweite Soundseite gezeigt wird. Im Schriftfeld werden unten zwei Striche angezeigt. '-' schaltet wieder zurück auf Seite 1 (ein Strich im Schriftfeld).
Durch die Taste 'M' und anschließend 'P' werden beide Sounds gemixt abgespielt. Nochmaliges Drücken von 'M' schält die Mixfunktion wieder aus. Diese Funktion wird durch ein 'M' im Schriftfenster dargestellt. Mit der Taste 'C' schließlich kann man beide Seiten löschen. Nach Betätigung dieser Taste erscheint ein '?' im Schriftfenster. Durch Drücken einer Taste außer '/' wird der Sound nicht gelöscht. Bei Druck auf '/' erscheint ein '!' im Schriftfenster und das Geräusch wird gelöscht.
Hier noch mal alle Funktionen im Überblick:

P = Abspielen
-> = Geschwindigkeit vergrößern
<- = Geschwindigkeit verringern
2 = Schrittweite für Geschwindigkeit auf 10 erhöhen
1 = Schrittweite für Geschwindigkeit auf 1 zurück
+ = Seite 2 zeigen und editieren
- = Seite 1 zeigen und editieren
M = Mixen
C = Sound löschen, Bestätigung mit '/'

Durch 'ESC' verläßt man den Editor und kommt zurück ins Menü.

4. ANHÖREN
Bei Wahl dieser Funktion muß eine Geschwindigkeit eingegeben werden. Mit dieser wird dann das im Speicher vorhandene Geräusch abgespielt.

5. DIGITALISIEREN
Über diese Funktion ist es möglich, in das Mikrofon eines am APPLE angeschlossenen und auf Aufnahme geschalteten Cassettenrecorder etwas hineinzusprechen / singen/ schreien... was dann im Speicher digital vorliegt und editiert werden kann.

6. FSC (FUNCTION SOUND CREATER)
Damit können Sinus-, Cosinus-, Tangenskurven usw. erzeugt werden und im Editor editiert werden. Es müssen lediglich ein x- und ein y-Wert angegeben werden. Dies ergibt oft die tollsten Weltraumsounds... Hier kann man sehr viel ausprobieren, die Ergebnisse sind oft sehr verblüffend.

7. MIXEN
2 Sounds werden von der Diskette geladen und können dann gemixt im Editor abgespielt werden.

Hinweise zur Programmeingabe
1. HEXDUMP NR.1 EINGEBEN UND ABSAVEN MIT 'BSAVE DSE.0B-J1,A$4000,L$150'
2. HEXDUMP NR.2 EINGEBEN UND ABSAVEN MIT 'BSAVE DES.0B-J2,A$4200,L1950'
3. BASIC STARTPROGAMM EINGEBEN UND MIT 'SAVE D-S-E.START' ABSPEICHERN
4. BASIC HAUPTPROGRAMM EINGEBEN UND MIT 'SAVE D-S-E' ABSPEICHERN
5. STARTEN DES DIGITAL SOUND EDITORS IMMER MIT 'RUN D-S-E.START'! (SEHR WICHTIG WEGEN VERLAGERUNG DES BASIC-STARTS)

Einbinden von mit DSE erzeugten Geräuschen in eigene Programme
1. Am Anfang des Programms müssen folgende Files geladen werden:
   a) die vorher mit DSE erzeugten und auf Disk abgespeicherten Files müssen folgendermaßen geladen werden:
      BLOAD (SOUND-NAME1),A$5000
      BLOAD (SOUND-NAME2),A$5100
      BLOAD (SOUND-NAME3),A$5200
      .
      .
      Bis zu maximal 16 Sounds können geladen werden
   c) Vom Basic Programm müssen dem Maschinenprogramm folgende Werte übergeben werden:
      Anfangsadresse des Sounds (POKE 253,A im Bereich von 80-94)
      Endadresse des Sounds (POKE 254, E Bereich von 81-95)
      Geschwindigkeit (POKE 255,G von 0-255)

Beispiel: Sound Nr. 2 soll mit Geschwindigkeit 10 abgespielt werden POKE 253,81: POKE 254,82: POKE 255,10
Der Aufruf erfolgt mit CALL 16640
Alles klar?
Wenn nicht, dann probieren Sie doch das Ganze mal durch, dann wird es Ihnen sehr schnell sehr einfach erscheinen ....

Zusätzliche Hinweise
Bei Verwendung von DSE-Files in eigenen Programmen darf HRG2 nicht benutzt werden! Hier werden nämlich die Sounds und das Maschinenprogramm abgespeichert.

BASIC-STARTPROGRAMM (ABSAVEN MIT 'SAVE D-S-E.START')

```
10  HOME : PRINT "D-S-E IS LOADING"
20  POKE 104,96: POKE 256 * 96,0
30  PRINT  CHR$ (4);"RUN D-S-E"
```

BASIC-HAUPTPROGRAMM (ABSAVEN MIT 'SAVE D-S-E')

```
10  REM ************************
15  REM ************************
20  REM **                    **
25  REM **   DIGITAL SOUND    **
30  REM **    E D I T O R     **
35  REM **                    **
40  REM **   WRITTEN 1985 BY  **
45  REM **     ROLAND GRAF    **
50  REM **                    **
55  REM **  D-S-E      (C) RGS **
60  REM **                    **
65  REM ************************
70  REM ************************
75  REM
80  ONERR  GOTO 10000
90  REM ^^ ERROR HANDLING ^^
100 :
101  REM > OBJ FILES LADEN <
105 :
110  PRINT  CHR$ (4);"BLOAD DSE.OBJ1,A$4000"
120  PRINT  CHR$ (4);"BLOAD DSE.OBJ2,A$4200"
121 :
130 :
131  REM * VARIABLEN *
135 :
140 DI = 16384:PL = 16640:SC = 1
150 ST = 18256:S = 80:E = 81:ST = 20480
160 G = 1:XC = 0:YC = 0:CR = 5:FG = 1:X = 266
170 A$(1) = "LADEN":A$(2) = "SAVEN":A$(3) = "EDITIEREN":A$(4) = "ANHOEREN":A$(5)
 = "DIGITALISIEREN":A$(6) = ">FSC<":A$(7) = "MIXEN":A$(8) = "ENDE"
180  POKE 253,S: POKE 254,E: POKE 255,G
190  POKE 232,0: POKE 233,66: SCALE= 1: ROT= 0
200 :
201  REM ** MENUE **
205 :
210  TEXT : HOME
220  INVERSE
230  PRINT "++++++++++++++++++++++++++++++++++++++++++++";: PRINT "++ DIGITAL SOUND
EDITOR  (C) RGS 1985 ++";: PRINT "++++++++++++++++++++++++++++++++++++++++++++";
240  NORMAL
250  FOR I = 1 TO 8
260  VTAB 4 + I: HTAB 10: PRINT A$(I)
```

```
270  NEXT
275  VTAB 14: HTAB 3: PRINT "RETURN = AUSFUEHREN": PRINT : HTAB 5: PRINT "PFEILT
ASTEN = AUSWAEHLEN"
280  VTAB CR: HTAB 10: INVERSE : PRINT A$(CR - 4): VTAB CR: HTAB 8: FLASH : PRIN
T CR - 4: VTAB CR: HTAB 11 +  LEN (A$(CR - 4)): PRINT CR - 4
290  IF  PEEK ( - 16384) > 127 THEN  GET IN$: GOTO 310
300  GOTO 290
305  VTAB 20: HTAB 1: PRINT CR
310  VTAB CR: HTAB 10: NORMAL : PRINT A$(CR - 4): VTAB CR: HTAB 8: PRINT " ": VT
AB CR: HTAB 11 +  LEN (A$(CR - 4)): PRINT " "
320  IF IN$ =  CHR$ (13) THEN 400
330  IF IN$ =  CHR$ (21) THEN CR = CR + 1: IF CR = 13 THEN CR = 5
340  IF IN$ =  CHR$ (8) THEN CR = CR - 1: IF CR = 4 THEN CR = 12
350  GOTO 280
400  ON CR - 4 GOTO 1000,2000,3000,4000,5000,6000,9000,7000
1000  REM ** LADEN **
1010  HOME
1020  PRINT ")))  SOUND LADEN ((("
1030  PRINT : PRINT "RETURN = CATALOG"
1040  PRINT : INPUT "FILENAME:";FI$
1050  IF FI$ = "" THEN  PRINT  CHR$ (4);"CATALOG": GOTO 1040
1055  IF SC = 1 THEN AS = 5000
1056  IF SC = 2 THEN AS = 5100
1060  PRINT  CHR$ (4);"BLOAD";FI$;",A$";AS
1070  GOTO 200
2000  REM ** SAVEN **
2010  HOME
2020  PRINT "(((  SOUND SAVEN  )))"
2030  PRINT : PRINT "RETURN = CATALOG"
2040  PRINT : INPUT "FILENAME:";FI$
2050  IF FI$ = "" THEN  PRINT  CHR$ (4);"CATALOG": GOTO 2040
2055  IF SC = 1 THEN AS = 5000
2056  IF SC = 2 THEN AS = 5100
2060  PRINT  CHR$ (4);"BSAVE";FI$;",A$";AS;",L$100"
2070  GOTO 200
3000  REM ** EDIT **
3005  POKE 255,G
3010  HGR : POKE  - 16302,0
3020  HCOLOR= 3: HPLOT 260,0 TO 279,0 TO 279,191 TO 260,191 TO 260,0
3021  IF SC = 1 THEN  DRAW 8 AT X,120
3022  IF SC = 2 THEN  DRAW 3 AT X,120
3023  IF FG = 10 THEN  HCOLOR= 3: DRAW 5 AT X,60
3024  IF M = 1 THEN  HCOLOR= 3: DRAW 46 AT X,100
3025  GOSUB 8000
3030  FOR I = 0 TO 191
3040  HPLOT  PEEK (ST + I),I TO 0,I
3050  NEXT
3060  IF  PEEK ( - 16384) > 127 THEN  GET IN$: GOTO 3500
3065 XC =  PEEK (ST + YC)
3070  XDRAW 4 AT XC + 1,YC:CX = XC + 1:CY = YC
3080  IF  PDL (0) > 200 THEN XC = XC + 1: IF XC = 256 THEN XC = 255
3090  IF  PDL (0) < 50 THEN XC = XC - 1: IF XC < 0 THEN XC = 0
3100  IF  PDL (1) > 200 THEN YC = YC + 1: IF YC = 192 THEN YC = 0
3110  IF  PDL (1) < 50 THEN YC = YC - 1: IF YC < 0 THEN YC = 191
3125  XDRAW 4 AT CX,CY
3127  IF  PDL (5) = 0 THEN 3060
3130  HCOLOR= 0: HPLOT 255,CY TO CX,CY
3140  HCOLOR= 3: HPLOT 0,YC TO XC,YC
3145  POKE ST + YC,XC
3150  GOTO 3060
```



```
3500  REM ** TASTE **
3510  IF IN$ =  CHR$ (21) THEN G = G + FG: GOSUB 8000
3520  IF IN$ =  CHR$ (8) THEN G = G - FG: GOSUB 8000
3530  IF IN$ = "P" THEN  HCOLOR= 3: DRAW 49 AT X,50: CALL PL: HCOLOR= 0: DRAW 49
 AT X,50
3540  IF IN$ =  CHR$ (27) THEN 200
3550  IF IN$ = "1" THEN FG = 1: HCOLOR= 0: DRAW 5 AT X,60
3560  IF IN$ = "2" THEN FG = 10: HCOLOR= 3: DRAW 5 AT X,60
3570  IF IN$ = "C" THEN  HCOLOR= 3: DRAW 32 AT X,70: GET IZ$: IF IZ$ <  > "/" TH
EN  HCOLOR= 0: DRAW 32 AT X,70: GOTO 3060
3580  IF IN$ = "C" THEN  HCOLOR= 0: DRAW 32 AT X,70: HCOLOR= 3: DRAW 2 AT X,70:
FOR I = 0 TO 255: POKE ST + I,0: NEXT : GOTO 3000
3590  IF IN$ = "M" AND M = 0 THEN M = 1: HCOLOR= 3: DRAW 46 AT X,100: POKE 253,S
: POKE 254,E + 1: GOTO 3060
3600  IF IN$ = "M" AND M = 1 THEN M = 0: HCOLOR= 0: DRAW 46 AT X,100: POKE 253,S
: POKE 254,E: GOTO 9500
3610  IF IN$ = ";" OR IN$ = "+" THEN ST = 20736:SC = 2: POKE 253,S + 1: POKE 254
,E + 1:M = 0: GOTO 3000
3620  IF IN$ = "-" THEN ST = 20480:SC = 1: POKE 253,S: POKE 254,E:M = 0: GOTO 30
00
3900  GOTO 3060
4000  REM ** HOEREN **
4010  HOME : PRINT "== AHHOEREN =="
4020  PRINT : INPUT "GESCHWINDIGKEIT:";G
4030  IF G < 0 OR G > 255 THEN 4010
4040  POKE 255,G: CALL PL
4050  PRINT : PRINT "NOCHMAL (J/N) ";: GET IN$: IF IN$ = "J" THEN 4010
4060  GOTO 200
5000  REM ** DIGIT **
5010  HOME : PRINT "## DIGITALISIEREN ##"
5020  PRINT : PRINT "GESCHWINDIGKEIT:";G
5030  PRINT : PRINT "ESC=EXIT"
5040  PRINT : PRINT "SPACE=START"
5050  GET IN$: IF IN$ =  CHR$ (27) THEN 200
5060  IF IN$ <  >" " THEN 5010
5070  PRINT : PRINT "DIGITALISIERUNG BEGINNT..."
5080  CALL DI
5090  PRINT : PRINT "FERTIG.": FOR W = 1 TO 300: NEXT W: GOTO 200
6000  HOME : PRINT "** FUNCTION SOUND CREATER **"
6005 CR = 7
6010  PRINT : PRINT "SIE KOENNEN WAEHLEN..."
6020  PRINT : PRINT "<1> SINUS"
6030  PRINT "<2> COSINUS"
6040  PRINT "<3> TANGENS"
6050  PRINT "<4> QUADRATWURZEL"
6060  PRINT "<5> QUADRAT"
6070  VTAB 11: HTAB 1: PRINT "< >": VTAB 11: HTAB 2: GET IN$: IF IN$ < "1" OR IN
$ > "5" THEN 6070
6080 F$(1) = "SINUS":F$(2) = "COSINUS":F$(3) = "TANGENS":F$(4) = "QUADRATWURZEL"
:F$(5) = "QUADRAT"
6090  HOME : PRINT F$( VAL (IN$))
6095 I =  VAL (IN$)
6100  ON I GOTO 6200,6350,6500,6650,6800
6200  REM ** SINUS **
6210  PRINT : INPUT "X-FAKTOR:";VG
6220  PRINT : INPUT "Y-FAKTOR:";NG
6300  FOR I = 0 TO 255
6310 PK = 128 + ( SIN (I / NG)) * VG: IF PK < 0 THEN PK = 0
6313  IF PK > 255 THEN PK = 255
6315  POKE ST + I,PK
```

```
6320  VTAB 7: HTAB 1: PRINT "< ";I;" >"
6330  NEXT
6340  GOTO 200
6350  REM ** COSINUS **
6360  PRINT : INPUT "X-FAKTOR:";VG
6370  PRINT : INPUT "Y-FAKTOR:";NG
6380  FOR I = 0 TO 255
6390 PK = 128 + ( COS (I / NG)) * VG: IF PK < 0 THEN PK = 0
6400  IF PK > 255 THEN PK = 255
6410  POKE ST + I,PK
6420  VTAB 7: HTAB 1: PRINT "< ";I;" >"
6430  NEXT
6440  GOTO 200
6500  REM ** TANGENS **
6510  PRINT : INPUT "X-FAKTOR:";VG
6520  PRINT : INPUT "Y-FAKTOR:";NG
6530  FOR I = 0 TO 255
6540 PK = 128 + ( TAN (I / NG)) * VG: IF PK < 0 THEN PK = 0
6550  IF PK > 255 THEN PK = 255
6560  POKE ST + I,PK
6570  VTAB 7: HTAB 1: PRINT "< ";I;" >"
6580  NEXT
6590  GOTO 200
6650  REM ** QUADRATWURZEL **
6660  PRINT : INPUT "X-FAKTOR:";VG
6670  PRINT : INPUT "Y-FAKTOR:";NG
6680  FOR I = 0 TO 255
6690 PK = 128 + ( SQR (I / NG)) * VG: IF PK < 0 THEN PK = 0
6700  IF PK > 255 THEN PK = 255
6710  POKE ST + I,PK
6720  VTAB 7: HTAB 1: PRINT "< ";I;" >"
6730  NEXT
6740  GOTO 200
6800  REM ** QUADRAT **
6810  PRINT : INPUT "X-FAKTOR:";VG
6820  PRINT : INPUT "Y-FAKTOR:";NG
6830  FOR I = 0 TO 255
6840 PK = 128 + (I ^ NG) * VG: IF PK < 0 THEN PK = 0
6850  IF PK > 255 THEN PK = 255
6860  POKE ST + I,PK
6870  VTAB 7: HTAB 1: PRINT "< ";I;" >"
6880  NEXT
6890  GOTO 200
6990  END
7000  REM ** ENDE **
7010  HOME : PRINT "PROGRAMM BEENDET. NEUSTART MIT 'RUN'"
7020  END
8000  REM ** DISPLAY **
8003  IF G > 255 THEN G = 0
8005  IF G < 0 THEN G = 255
8010 X = 266:Y1 = 12:Y2 = 20:Y3 = 28
8020  IF G < 10 THEN G$ = "00" +  STR$ (G)
8030  IF G < 100 THEN G$ = "0" +  STR$ (G)
8040  IF G >  = 100 THEN G$ =  STR$ (G)
8050 Z1 =  VAL ( LEFT$ (G$,1))
8060 Z2 =  VAL ( LEFT$ ( RIGHT$ (G$,2),1))
8070 Z3 =  VAL ( RIGHT$ (G$,1))
8080  HCOLOR= 0: FOR L = 266 TO 275: HPLOT L,1 TO L,34: NEXT
8090  HCOLOR= 3: DRAW Z1 + 17 AT X,Y1: DRAW Z2 + 17 AT X,Y2: DRAW Z3 + 17 AT X,Y
3
```



```
8095  POKE 255,G
8100  RETURN
9000  REM ** MIXEN **
9005 M = 1
9010  HOME : PRINT ">> DSE SPECIAL MIX <<"
9015  PRINT : PRINT "CATALOG (J/N)";: GET IN$: IF IN$ = "J" THEN  PRINT  CHR$ (4
); CHR$ (4);"CATALOG"
9020  PRINT : INPUT "FILENAME 1 : ";F1$
9030  PRINT : INPUT "FILENAME 2 : ";F2$
9040  PRINT  CHR$ (4);"BLOAD";F1$;",A$5000"
9050  PRINT  CHR$ (4);"BLOAD";F2$;",A$5100"
9060  POKE 253,S: POKE 254,E + 1
9065 M = 1
9070  GOTO 200
9500  REM ** PAGES **
9510  IF SC = 1 THEN  POKE 253,S: POKE 254,E
9520  IF SC = 2 THEN  POKE 253,S + 1: POKE 254,E + 1
9600  GOTO 3060
10000  REM ** ERROR **
10010  TEXT : HOME : PRINT "FEHLER...<"; PEEK (222);">": GET A$: GOTO 200
```

Fortsetzung Seite 67

Eine leistungsfähige Hilfe für den Assembler-Programmierer

„Disassembler“ für den VC-20-Computer

Das Programm „Disassembler“ ist in Basic geschrieben und ca. 2700 Bytes lang. Wie der Name schon sagt disassembliert es ein Maschinensprachprogramm und gibt das Ergebnis auf einen 80-Zeichen-Drucker aus. Da druckerspezifische Befehle weggelassen wurden, läuft das Programm auf jedem Drucker. Kern des Programms sind die Datazeilen 10-36. Sie enthalten jeden Befehl des 6502 (6510) und die jeweilige Adressierungsart. Die Zahl hinter dem Mnemonic gibt die Adressierungsart an. Sie wird in der Zeile 113 vom Rest des Strings (h§) getrennt und durch die Zeilen 114 und 3000-3013 der jeweiligen Darstellung zugeordnet.

Die Zellen 51-57 enthalten den Kopf der Ausgabe. Für einen geordneten, tabellarischen Ausdruck sorgen die Zeilen 101, 103 und 3012 in Verbindung mit 3500 und 3501.

Da die Ausgabe jeweils in dezimaler Form und hexadezimaler Form erfolgt, wandeln die Zeilen 5000-5015 eine dezimale Zahl in eine hexadezimale um. Bei einer relativen Adressierung wird automatisch die Zieladresse berechnet und ausgegeben. Dafür sorgen die Zeilen 2990-2993.

Damit man sich ein besseres Bild vom Programm machen kann, hat der Autor ein zusammenhangloses Programm disassembliert, das jede Adressierungsart durch ein Beispiel darstellt.

Anwendung:

Wenn man ein Maschinensprachprogramm, das sich im Speicher befindet, disassemblieren will, so muß man es erst vor Überschreibung schützen. Dazu setzt man den Basicanfang herauf (für Programme, die am Anfang stehen) bzw. das Ende herunter. Falls das Programm außerhalb des Rams liegt, ist dies nicht nötig.

Das Heraufsetzen des Basicanfangs geschieht durch folgende Zeile:

POKE43, Zieladresse AND 255: POKE 44, Zadr / 256: POKE Zadr -1,0: NEW

Heruntersetzen:

POKE55, Zadr AND255:POKE56, Zadr / 256:CLR

Danach lädt man den „Disassembler“, gibt den Namen des M-Programms und die Anfangs- und Endadresse ein.

Der Drucker eignet sich als Ausgabegerät für ein Maschinenprogramm mehr als der Bildschirm, weil es zu viele Informationen sind, die dargestellt werden müssen und man deshalb leicht den Überblick verliert.

```
0 POKE36879,26:PRINT"■";:CLR
1 PRINT"**********************";
2 PRINT"*   CARSTEN BIEKER    *";
3 PRINT"* V.BRUCHHAUSENSTR.5A*";
4 PRINT"* 4350 RECKLINGHAUSEN*";
5 PRINT"*  02361 / 16117      *";
6 PRINT"**********************";
```

```
10 DATABRK,ORA4,?,?,?,ORA2,ASL2,?,PHP,ORA1
11 DATAASL,?,?,ORA8,ASL8
12 DATA?,BPL11,ORA5,?,?,?,ORA3,ASL3,?,CLC
13 DATAORA10,?,?,?,ORA9,ASL9,?,JSR8,AND4
14 DATA?,?,BIT2,AND2,ROL2,?,PLP,AND1,ROL,?
15 DATABIT8,AND8,ROL8,?,BMI11,AND5
16 DATA?,?,?,AND3,ROL3,?,SEC,AND10,?,?
17 DATA?,AND9,ROL9,?,RTI,EOR4,?,?,?,EOR2
18 DATALSR2,?,PHA,EOR1,LSR,?,JMP8,EOR8,LSR8,?
19 DATABVC11,EOR5,?,?,?,EOR3,LSR3,?,CLI,EOR10
20 DATA?,?,?,EOR9,LSR9,?,RTS,ADC4,?,?
21 DATA?,ADC2,ROR2,?,PLA,ADC1,ROR,?,JMP12,ADC8
22 DATAROR8,?,BVS11,ADC5,?,?,?,ADC3,ROR3,?
23 DATASEI,ADC10,?,?,?,ADC9,ROR9,?,?,STA4
24 DATA?,?,STY2,STA2,STX2,?,DEY,?,TXA,?
25 DATASTY8,STA8,STX8,?,BCC11,STA5,?,?,STY3,STA3,STX6
26 DATA?,TYA,STA10,TXS,?,?,STA9,?,?,LDY1
27 DATALDA4,LDX1,?,LDY2,LDA2,LDX2,?,TAY,LDA1
28 DATATAX,?,LDY8,LDA8,LDX8,?,BCS11,LDA5,?,?
29 DATALDY3,LDA3,LDX6,?,CLV,LDA10,TSX,?,LDY9,LDA9
30 DATALDX10,?,CPY1,CMP4,?,?,CPY2,CMP2,DEC2,?
31 DATAINY,CMP1,DEX,?,CPY8,CMP8,DEC8,?,BNE11
32 DATACMP5,?,?,?,CMP3,DEC3,?,CLD,CMP10,?
33 DATA?,?,CMP9,DEC9,?,CPX1,SBC4,?,?,CPX2,SBC2
34 DATAINC2,?,INX,SBC1,NOP,?,CPX8,SBC8,INC8,?
35 DATABEQ11,SBC5,?,?,?,SBC3,INC3,?,SED,SBC10
36 DATA?,?,?,SBC9,INC9,?
50 OPEN1,4
51 PRINT"▓▓NAME DES PROGRAMMS":INPUTA$:PRINT#1,"                                   PROGRAMM "A
$
52 A$=" SPEICHERZELLE|  INHALT      (HEX.)  |  BEDEUTUNG     |         BEMERKUNGEN"
53 PRINT#1,A$
54 A$="  HEX.  DEZ.  |  + 0   + 1   + 2  |  MNEMONICS   |"
55 PRINT#1,A$
56 A$="────────────────┼─────────────────┼───────────────┼──────────────────────
"
57 A$=A$+"───":PRINT#1,A$
60 PRINT"▓▓▓1./LETZTE SPEICHERZELLE":INPUTB,S:D=B:GOSUB5000:B$=H$:GOTO100
99 B=B+1:D=B:GOSUB5000:B$=H$:IFB>STHEN10000
100 A$="  "
101 C=6:GOSUB3500
102 A$=A$+B$:B$=STR$(B)
103 C=12:GOSUB3500
104 A$=A$+B$+"  |  `"
110 M=PEEK(B):D=M:GOSUB5000:M$=RIGHT$(H$,2):N=PEEK(B+1):D=N:GOSUB5000:N$=RIGHT$(
H$,2)
111 O=PEEK(B+2):D=O:GOSUB5000:O$=RIGHT$(H$,2)
112 FORI=OTOM:READH$:NEXT:RESTORE
113 P$=LEFT$(H$,3):P=VAL(MID$(H$,4,2))+1
114 ONPGOTO3000,3001,3002,3003,3004,3005,3006,3007,3008,3009,3010,2990,2995
2990 IFN<128THEN2992
2991 N=N-256
2992 P$=P$+" ->"+STR$(B+2+N)
2993 GOTO3007
2995 P$=P$+"("+STR$(N+O*256)+")"
2996 GOTO3011
3000 A$=A$+M$+"                   |   "+P$:GOTO3012
3001 P$=P$+" # "+STR$(N):GOTO3007
3002 P$=P$+" $ "+STR$(N):GOTO3007
3003 P$=P$+" $ "+STR$(N)+",X":GOTO3007
3004 P$=P$+" ("+STR$(N)+",X)":GOTO3007
3005 P$=P$+" ("+STR$(N)+"),Y":GOTO3007
3006 P$=P$+" $ "+STR$(N)+",Y"
3007 A$=A$+M$+"    "+N$+"              |  "+P$:B=B+1:GOTO3012
3008 P$=P$+" $ "+STR$(N+256*O):GOTO3011
3009 P$=P$+" $ "+STR$(N+256*O)+",X":GOTO3011
3010 P$=P$+" $ "+STR$(N+256*O)+",Y"
3011 A$=A$+M$+"    "+N$+"    "+O$+"    |  "+P$:B=B+2
3012 B$="":C=51:GOSUB3500
3013 A$=A$+"|":PRINT#1,A$:GOTO99
3500 IFLEN(A$)+LEN(B$)<CTHENA$=A$+" ":GOTO3500
3501 RETURN
5000 F=INT(D/256):GOSUB5010
5001 X$=H$
5002 F=D-F*256:GOSUB5010:H$=X$+H$:RETURN
5010 L=FAND15:H=(FAND240)/16
5011 IFH>9THENH=H+55:GOTO5013
5012 H=H+48
5013 IFL>9THENL=L+55:GOTO5015
5014 L=L+48
5015 H$=CHR$(H)+CHR$(L):RETURN
10000 A$="-------- * ENDE * --------":PRINT#1,A$:CLOSE1:END
READY.
```


ERROR-HELPER für den Atari 800 XL

Dieses Programm läuft nur auf Diskette

FUNKTION UND AUFBAU:
Eine Schattenseite des Atari's sind seine Fehlermeldungen. Tritt ein Fehler auf, meldet er schlicht „ERROR X IN LINE Y". Das Programm „Error-Helper" schafft diesem Dilemma Abhilfe. Es springt nach einem aufgetretenen Fehler in ein Unterprogramm, welches die normale Fehlermeldung, zusätzlich aber noch einige Hilfen von der Diskette liest und auf dem Bildschirm ausgibt. Danach kann man aus einem Menü noch auswählen, ob man z. B. die Fehlerzeile listen oder einfach übergehen und mit der darauffolgenden Zeile fortfahren will.

EINGABE:
Die Eingabe des Programms erfolgt in 2 Stufen. Als erstes muß man natürlich auf eine Diskette die Texte, die man später abrufen will, schreiben. Dieses erledigt Listing 1 für Sie. Man muß aber keineswegs alle Fehlermeldungen abtippen. Besitzt man z. B. kein Interface-Modul, braucht man die Meldungen, die sich auf dieses beziehen, nicht abzutippen. Die Diskette, auf die die Fehlermeldungen geschrieben werden sollen, darf keine anderen Programme mehr beinhalten, da „Error-Helper" die gesamte Directory belegt! Ist die „Error-Helper"-Masterdiskette geschrieben, speichern Sie das Listing 1 auch auf die Diskette.
Als nächstes speichern Sie Listing 2 auf die Diskette. Es erledigt die Aufgabe, den aufgetretenen Fehler anzuzeigen und die Hilfen von Diskette zu laden. Bevor Sie nun ein Programm in ihren Atari eingeben wollen, laden Sie Listing 2, und schon ist die Fehlerbehandlung kinderleicht.

WAS SIE BEACHTEN MÜSSEN:
„Error-Helper" setzt in Zeile 0 die TRAP Funktion des Atari Basics auf 30000. Diesen Wert darf das Programm, das eingegeben wird, nicht verändern. Braucht das Programm jedoch auch die TRAP Funktion, dann weisen Sie einer Variablen mit Namen Trap die Zeile zu, in die gesprungen werden soll. Zum Beispiel:

NORMALES-LISTING
10 TRAP 100 : OPEN "1,4,0, "D:"
UMGEÄNDERTES LISTING
10 LET = 100 : OPEN "1, 4, 0, "D:"

Das X Programm springt nach einem vorausgesehenen Fehler zunächst in die BASIC-Trapzeile, bekommt dort jedoch die Anweisung, in die Zeile der Variablen Trap zu springen.
Außerdem darf das einzugebende Programm weder Zeile 0, noch die Zeilen ab 30000 überschreiben, da dort „Error-Helper" gespeichert ist.

VARIABLEN FÜR LISTING 1:
NUMMER § = Fehlernummer, die gerade abgespeichert wird
FILE § = Name des Files, welches abgespeichert wird
TEXT § = Text, der abgespeichert wird.

VARIABLEN FÜR LISTING 2:
NUMMER § = Fehlernummer
FILE § = Name des zu ladenden Files
LESES § = Input des Textes
TRAP = Zeile, zu der gesprungen werden soll
TRAP 2 = Hilfe zu TRAP
DLHELP = Anfangsadresse der Displaylist
ZEILE = Fehlerzeile
TASTE = Taste gedrückt?
NZEILE = Adresse der nächsten Zeile
ADRESSE = Berechnung der nächsten Zeile
BZEILE = Berechnung der nächsten Zeile

Atari-„Error-Helper“: Schluß mit den Fehlercodes – endlich Fehlermeldung im Klartext!

```
0 REM ERROR-HELPER MASTERDISC WRITER
1 REM
2 REM GUIDO MICHEL   TEL.0211/151585
3 REM FLEHER STR.187
4 REM 4000 DUESSELDORF 1
5 REM
10 DIM NUMMER$(5),FILE$(15),TEXT$(255):? "     SCHREIBE ERROR-HELPER MASTERDISK"
:TRAP 50
20 FILE$="D:FEHLER."
30 READ NUMMER$,TEXT$:FILE$(LEN(FILE$)+1)=NUMMER$
40 OPEN #1,8,0,FILE$:PRINT #1;TEXT$:CLOSE #1:GOTO 20
50 END
60 DATA 2,Nicht genug Speicherplatz vor handen; oder zu viele FOR-NEXT Schleifen
 bzw. Unterprogramme verschachtelt
70 DATA 3,Ein numerischer Wert ist zu gross zu  klein oder an einer Stelle negat
iv wo er positiv sein muss
80 DATA 4,In einem Programm wurden mehr als 128 zugelassene Variablen benuzt
90 DATA 5,Es ist ein Teil einer alphanumerischenVariablen angesprochen worden fu
er densie nicht dimensioniert war
100 DATA 6,In einem Programm sollten mehr Daten  gelesen werden als vorhanden si
nd
110 DATA 7,Ein numerischer Wert ist groesser als 32767 bzw. kleiner als 0
120 DATA 8,Es wurde bei einem INPUT festgestellt dass der gelesenen Wert nicht z
ur zu- gewiesenen Variable passen
130 DATA 9,Eine Variable wurde bereits dimensio- niert oder sie soll mehr als 32
767    Bytes belegen
140 DATA 10,Ein Ausdruck enthaelt zu Viele Klam-  mern bzw. zu viele verscachtel
te Funk-tionen
150 DATA 11,Es wurde versucht durch 0 zu teilen   oder eine Zahl ist groesser bz
w. klei-ner als 9.99999999*10^97
160 DATA 12,Bei einem Sprungbefehl soll zu einer  Zeile gesprungen werden die ni
cht vor-handen ist
170 DATA 13,Es wurde ein NEXT Befehl ohne FOR     Anweisung gefunden(Untersuche
Programmnach POP Anweisungen)
180 DATA 14,Ein Befehl ist zu komplex oder geht   ueber das Ende einer logischen
 Zeile  hinaus
190 DATA 15,Nach einem RETURN bzw.NEXT konnte die Zeile mit GOSUB bzw.FOR nicht
mehrge- funden werden
200 DATA 16,Es ist ein RETURN ohne ein vorrange-  gangenes GOSUB gefunden worden
210 DATA 17,Ein Befehl ist durch POKE so veraen-  dert worden dass er nicht mehr
 erkanntwerden kann
220 DATA 18,Eine Variable soll mit Hilfe von VAL  in eine numerische umgewandelt
 werden obwohl dies nicht geht
230 DATA 19,Es wurde versucht ein Programm zu la- den dass den Speicher des Comp
uters   ueberschteitet
240 DATA 20,Es wurde versucht Kanal 0 oder einen  Kanal ueber 7 zu benutzen
250 DATA 21,Es soll ein mit LIST abgespeichertes  Programm mit LOAD geladen werd
en
260 DATA 128,Waehrend einer Ein-Ausgabeoperation   wurde die BREAK-Taste gedruec
kt
270 DATA 129,Es wurde versucht einen bereits offe- nen Kanal erneut zu oeffnen
280 DATA 130,Es wurde versucht auf ein unbekanntes Geraet zurueckzugreifen
290 DATA 131,Der angesprochene Kanal ist nur fuer  Ausgabe geoeffnet
300 DATA 132,Bei der Ausfuehrung eines XIO-Befehls ist ein Fehler aufgetreten
310 DATA 133,Es wurde ein nicht geoeffneter Kanal  angesprochen
320 DATA 134,Angegeben Kanalnummer ist kleiner 1   bzw. groesser 7
330 DATA 135,Der angesprochene Kanal ist nur fuer  Eingabe geoeffnet
340 DATA 136,Es wurde ein EOF-Record gelesen oder  ein Sektor soll gelesen werde
n   der  nicht geoeffnet war
350 DATA 137,Der Computer konnte einen Rekord nichtvollstaendig empfangen da er
groesser als 256 Bytes war
360 DATA 138,Das angegebene Geraet konnte nicht an-gesprochen werden (Geraet ein
geschal- tet;Schalter auf ONLINE?)
370 DATA 139,Der Programmrecorder die Diskettensta-tion oder das Interfacemodul
arbeiten nicht einwandfrei
380 DATA 140,Bei der Datenuebertragung ueber den   seriellen Bus ist ein Fehler
aufgetre-ten.(Disk/Cass defekt ?)
390 DATA 141,Der Curser wurde auserhalb der zuge-  lassenen Bildbegrenzung gesez
t
400 DATA 142,Bei der Datenuebertraging ueber den seriellen Bus ist ein Fehler au
fgetre-   ten.(Disk/Cass defekt ?)
410 DATA 143,Pruefsumenfehler bei der Uebertragung ueber den seriellen Bus(Disk/
Cass de- fekt ?)
420 DATA 144,Eine Diskette ist schreibgeschuetzt   oder in der Directory befinde
t sich   ein Fehler
430 DATA 145,Beim Verify der Disk wurde ein Unter- schied festgestellt oder Fehl
erbei derBildschirmdarstellung
440 DATA 146,Die verlangte Funktion ist nicht aus- fuehrbar(z.B.Daten von der Ta
statur zulesen u.s.w)
450 DATA 150,Es kann nur 1 Ein/Ausgabekanal fuer   eine best.serielle Schnittste
lle ge-   oeffnet sein
460 DATA 151,Schnittstelle wurde nicht durch einen XIO 40-Befehl fuer gleichzeit
ige  Ein/Ausgabe geoeffnet
470 DATA 152,Die bezueglich des Buffers gemachten  Angaben waren nicht konkret g
enug
480 DATA 153,Ueber eine seri.Schnitt.sollten Daten ein/ausgegeben werden obwohl
eine andere in XIO40 Betrieb ist
490 DATA 154,Obwohl durch eine Operation verlangt  war die Schnittstelle nicht f
uer Ein- & Ausgabe aktiviert
500 DATA 160,Es koennen fuer die Diskettenstationennur die Geraetenamen D: D1: D
2: D3: oder D4: verwendet werden
510 DATA 161,Es sind zu viele File gleichzeitig ge-oeffnet (je nach DOS verschie
den viele Files moeglich)
520 DATA 162,Auf der Diskette ist kein Speicher-   platz mehr vorhanden
530 DATA 163,Waehrend einer Ein/Ausgabeoperation   wurde ein nicht zu behebender
 unbekannter Fehler festgestellt
540 DATA 164,Die Verbindung zwischen den einzelnen sektoren sind zerstoert oder
Zeiger   auserhalb des Files
550 DATA 165,Ein Filename enthaelt unzulaessige Zeichen oder die Wild-Card-Zeich
en wurdennicht richtig benuzt
560 DATA 166,Das bei einem POINT Befehl angegebene Byte existiert nicht
570 DATA 167,File ist schreibgeschuetzt
580 DATA 168,Es wurde versucht einen nicht existie-renden XIO-Befehl auszufuehre
n
590 DATA 169,Es ist kein Platz mehr in der Direc-  tory.Es koennen nur 64 Namen
gespei   chert werden
600 DATA 170,Das angegebene File ist nicht auf der Diskette
610 DATA 171,Es wurde mit Hilfe eines POINT-Befehlsversucht auf einen sektor zuz
ugreifen der nicht geoeffnet ist
```

Basic-Erweiterung für den Spectrum

Nützliche Befehlerweiterung des Spectrum-Editors

Das hier vorgestellte Maschinenprogramm erweitert das Basic des Spectrum um einige nützliche Befehle für die Eingabe und Bearbeitung von Basic-Programmen. Das Programm läuft auf dem Spectrum mit 16K und mit 48K RAM. Vorhandene Maschinenprogramme, die einen festen Platz im Speicher belegen, werden durch dieses Programm nicht überschrieben.
Im einzelnen stehen folgende Befehle zur Verfügung:
-AUTO-Funktion einschalten
(*a <1. Zeilennummer>,<Abstand><Enter>)
Die AUTO-Funktion erleichtert das Eingeben von Basic-Programmen: die Zeilennummern müssen nicht mehr eingetippt werden, sondern werden automatisch vorgegeben. Nach der Eingabe des Befehls erscheint die angegebene erste Zeilennummer im Editorbereich des Bildschirms. Die Basic-Zeile kann nun eingegeben werden, ohne daß die Eingabe der Zeilennummer erforderlich ist. Nach Abschluß der korrekten Eingabe mit <ENTER> erscheint die nächste Zeilennummer, entsprechend dem gewählten Abstand. Dies wird immer weiter fortgesetzt. Um die AUTO-Funktion auszuschalten, ist es notwendig, die vorgegebene Zeilennummer zu löschen und nur <ENTER> einzugeben. Die AUTO-Funktion wird automatisch ausgeschaltet, wenn die vorzugebende Zeilennummer größer als 9999 ist.
Existiert zu einer vorgegebenen Zeilennummer bereits eine Zeile im Programm, so erscheint vor der Zeilennummer das Copyright-Zeichen. Durch Eingabe von <EDIT> (<CAPS SHIFT><1>) kann diese in den Editor übernommen werden. Es kann natürlich auch eine neue Zeile eingegeben werden, die dann die alte ersetzt. Wird hinter der Zeilennummer nur <ENTER> eingegeben, so wird die alte Zeile im Programm gelöscht.
Wird die vorgegebene Zeilennummer geändert, so wird die eingegebene Zeile unter der geänderten Zeilennummer abgespeichert. Die Vorgabe der nächsten Zeilennummer wird von dieser Änderung nicht beeinflußt.
Wird bei der Eingabe des Befehls kein Abstand angegeben (*a <1. Zeilennummer><ENTER>), so wird zwischen den Zeilennummern automatisch ein Abstand von 10 eingesetzt. Wird auch keine Zeilennummer angegeben (*a<ENTER>), so wird als erste Zeilennummer 10 und als Abstand ebenfalls 10 eingesetzt.
-Editieren einer bestimmten Zeile
(*e <Zeilennummer><ENTER>)
Die angegebene Zeile wird zum Editieren in den Editorspeicher gebracht und angezeigt. Existiert die angegebene Zeile nicht im Programm, so wird „Nonsense in Basic“ gemeldet. Mit diesem Befehl können Zeilen editiert werden, ohne daß erst der Programm-Cursor innerhalb des Listings dorthin gebracht werden muß.
-Löschen von Zeilen
(*d <1. Zeilennummer>,<2. Zeilennummer><ENTER>)
Alle Zeilen zwischen der ersten und der zweiten angegebenen Zeile (einschließlich dieser beiden) werden gelöscht. Die angegebenen Zeilen müssen vorhanden sein, sonst erfolgt die Meldung „Nonsense in Basic“. Wird die zweite Zeilennummer weggelassen (*d <1. Zeilennummer>,<ENTER>), so wird ab der ersten Zeilennummer bis zum Ende des Basic-Programms gelöscht. Wird die erste Zeilennummer weggelassen (*d,<2. Zeilennummer ><ENTER>), so werden alle Zeilen vom Anfang des Basic-Programms bis einschließlich der angegebenen Zeile gelöscht. Werden beide Zeilennummern weggelassen (*d<ENTER>), so wird das gesamte Basic-Programm gelöscht. Wird nur eine Zeilennummer angegeben (*d <Zeilennummer><ENTER>), so wird die angegebene Zeile gelöscht.
-Listen von Zeilen
(*l<1. Zeilennummer>,<2. Zeilennummer><ENTER>
für Bildschirm, *p<1. Zeilennummer>,<2. Zeilennummer><ENTER> für Drucker)
Alle Zeilen zwischen der ersten und der zweiten angegebenen Zeile werden auf dem Bildschirm bzw. dem Drucker aufgelistet. Die angegebenen Zeilen müssen nicht im Programm vorhanden sein. Wird die zweite Zeilennummer weggelassen (*l <1. Zeilennummer>,<ENTER>), so werden die Zeilen ab der angegebenen Zeilennummer bis zum Ende des Programms gelistet (entspricht LIST <1. Zeilennummer>). Wird die erste Zeilennummer weggelassen (*l ,<2. Zeilennummer><ENTER>), so werden alle Zeilen vom Anfang des Basic-Programms bis zur angegebenen Zeile gelistet. Werden beide Zeilennummern weggelassen (*l <ENTER>), so wird das gesamte Basic-Programm gelistet (entspricht LIST). Wird nur eine Zeilennummer angegeben (*l <Zeilennummer><ENTER>), so wird nur die angegebene Zeile gelistet. (Was oben für *l angeführt wurde, gilt natürlich auch für *p.)
Die Ausgabe auf den Drucker erfolgt über den Druckerkanal. Es wird also die gleiche Druckerroutine wie für LLIST und LPRINT benutzt.
-Kopieren einer Zeile
(*c <Zeilennummer>,<neue Zeilennummer><ENTER>)
Eine Kopie der als erstes angegebenen Zeile wird angefertigt und mit der neuen Zeilennummer versehen ins Programm eingefügt. Die zu kopierende Zeile muß vorhanden sein, sonst erfolgt die Meldung „Nonsense in Basic“. Ist im Programm bereits eine Zeile mit der neuen Zeilennummer vorhanden, so wird diese gelöscht.
-Verschieben einer Zeile
(*m <alte Zeilennummer>,<neue Zeilennummer><ENTER>)
Der Befehl entspricht dem Befehl zum Kopieren einer Zeile, zusätzlich wird hier die ursprüngliche Zeile aus dem Programm gelöscht. Referenzen (GOTOs, etc.) auf die ursprüngliche Zeile werden nicht verändert.
-Anwählen der ersten bzw. letzten Zeile des Basic-Programms
(*<ENTER> bzw. **<ENTER>)
Der Programm-Cursor wird auf die erste bzw. letzte im Programm vorhandene Zeile gesetzt und ein automatisches Listing angezeigt.
Alle Befehle können nur als direkte Befehle, nicht innerhalb einer Programmzeile verwendet werden. Es kann immer nur ein Befehl angegeben werden. Die Befehlsbuchstaben können kleine oder große Buchstaben sein. Die Zeilennummern müssen direkt angegeben werden, nicht als Variablen oder Ausdrücke. Sie müssen zwischen 1 und 9999 liegen. Zwischen Befehlsbuchstaben und Zeilennummer sowie zwischen Komma und Zeilennummer sind beliebig viele Leerzeichen zulässig.
Das Maschinenprogramm „xbasic“ wird mit LOAD „xbasic“CODE : RANDOMIZE USR 16384 in den Bildschirmspeicher geladen und gestartet. Diese beiden Befehle müssen direkt hintereinander ausgeführt werden. Durch dieses Maschinenprogramm wird RAMTOP um 859 Bytes herabgesetzt, um Platz für das Erweiterungsprogramm zu schaffen. Ist dies nicht möglich, so erscheint die Meldung „RAMTOP no good“. Sonst aber werden alle absoluten Adressen der Erweiterung umgerechnet und die Erweiterung oberhalb des neuen RAMTOP plaziert. Bereits vorhandene Maschinenprogramme bleiben also unversehrt erhalten. Es werden jedoch alle Variablen und der GOSUB-Stack gelöscht (wie bei CLEAR). Nachdem sich die Erweiterung im Speicher befindet, wird im unteren Teil des Bildschirms die Startadresse der Erweiterung angezeigt. Sie bleibt solange stehen, bis eine Taste gedrückt wird. Die Anzeige der Startadresse hat zwei Gründe: zum einen ist diese Startadresse ein Hinweis darauf, wo RAMTOP jetzt steht (Startadresse-1). Zum anderen ist es über diese Startadresse möglich (durch (RANDOMIZE USR Startadresse), die Erweiterung wieder zu aktivieren, wenn sie (etwa durch NEW) abgeschaltet wurde. Solange die Erweiterung aktiv ist, wird vor jede Fehler- oder OK-Meldung das Zeichen „#“ gesetzt.

```
  1 REM xbasic
  2 REM
  3 REM Basic-Erweiterung
  4 REM fuer den Spectrum
  5 REM
  6 REM Gerd Kluge
  7 REM Augustastr. 7
  8 REM 5810 Witten
  9 REM
100 REM Maschinenprogramm in
110 REM den Bildschirmspeicher
120 REM POKEn
130 LET start=16384: LET n=1085
140 LET z=0: LET adr=start
150 READ z$,check: LET z=z+1
160 LET p=1: LET summe=0
170 IF LEN z$/2<>INT (LEN z$/2)
 THEN GO TO 300
180 LET hi=CODE z$(p)-48-7*(z$(
p)>="A")
190 LET lo=CODE z$(p+1)-48-7*(z
$(p+1)>="A")
200 LET byte=lo+16*hi
210 POKE adr,byte
220 LET summe=summe+byte
230 LET p=p+2: LET adr=adr+1
240 IF p<LEN z$ THEN GO TO 280
250 IF summe<>check THEN GO TO
310
260 IF adr<start+n THEN GO TO 1
50
270 GO TO 400
280 IF adr<start+n THEN GO TO 1
80
290 PRINT "Zu viele Bytes";: GO
 TO 320
300 PRINT "Falsche Anzahl Zeich
en";: GO TO 320
310 PRINT "Pruefsummenfehler";
320 PRINT " in Zeile ";10*z+490
: STOP
370 REM
380 REM Maschinenprogramm
390 REM auf Band sichern
400 SAVE "xbasic"CODE start,n
410 INPUT "Nochmal (j/n) ? ";i$
420 IF i$="" THEN GO TO 410
430 IF i$="j" THEN GO TO 400
440 IF i$="J" THEN GO TO 400
470 REM
480 REM 1085 Bytes
490 REM Maschinenprogramm
500 DATA "ED5B4B5C2A595C2BCDE51
9CDBF162AB25C115B03AFED52E52A655
C11320019D1",3320
510 DATA "ED5230072AB45CED52300
2CF15ED53B25C2A3D5C4E2346EB363E2
BF9C5ED733D",3506
520 DATA "5C2AB25C2311E240AFED5
2444DDD219240DD6E00DD23DD6600DD2
37DB4280B5E",3465
530 DATA "2356EB09EB722B7318E7E
D5BB25C1321E240015B03EDB03EFDCD0
116ED4BB25C",3700
540 DATA "03CD2B2DCDE32D010000C
D3D1F2AB25C23E906418141884 1B941C
441E741F541",3234
550 DATA "FB41004205420E4213422
8423C423F426842C242EA422A4335437
C439E43AA43",2591
560 DATA "BC43D643F243F5430844B
141C642A743B541D242E7421D438343A
242BE429343",3784
570 DATA "0000ED7B3D5CD176FDCB0
1AEFDCB304EC4CD0E3A3A5C3CF521000
0FD7437FD74",3807
580 DATA "26220B5CAF32344423221
65CCDB016FDCB37AECD6E0D3E23D73E2
0D7FDCB02EE",3436
590 DATA "F147FE0A3802C607CDEF1
53E20D778119113CD0A0CAF113615CD0
A0CED4B455C",3108
600 DATA "CD1B1A3E3AD7FD4E0D060
0CD1B1ACD97103A3A5C3C281BFE09280
4FE152003FD",2778
610 DATA "340D01030011705C21445
```

```
CCB7E280109EDB8FD360AFFFDCB019E1
84AFD363102",2926
 620 DATA "3A3444B7F528062A35442
2495CCD9517CDB016F128303EFFCD011
6ED4B495CC5",3342
 630 DATA "CD1B1AE1CD6E192007AFC
D01163E7FD72A3544ED4B37440922354
4EB21F0D819",3180
 640 DATA "3004AF323444AFCD0116C
D2C0F2A595C225D5CDFFE2A204221000
022455C3E01",2414
 650 DATA "32475CE7FE0DCA6E42FE2
A287DCD8D2C3028CBAF11E740D5FE44C
AA643FE4CCA",4220
 660 DATA "E043FE50CAE343FE45287
2FE43CAC842FE4DCAC842FE41287DD1C
D171BFDCB00",4590
 670 DATA "7E2013FDCB3066CAE7402
A595CCDA711FD3600FF1890CDFB1978B
1C22C433234",3807
 680 DATA "44DFFE0D2835FDCB3046C
4AF0DCD6E0D3E19FD964F328C5CFDCB0
1FEFD3600FF",4061
 690 DATA "FD360A01CD8A1BC3E7402
10F273E210000CD6E19CD9516ED53495
C1848CD021A",2890
 700 DATA "DFFE0D2026C5E1CD6E192
01FED43495CCDA90FF1189ECD021A78B
12003010A00",3242
 710 DATA "ED433544DFFE0D2810FE2
C2006CD021ADFFE0D206D78B12003010
A00ED433744",2941
 720 DATA "3C323444F1C37C41F5CD0
21A78B12852ED433944DFFE2C2049CD0
21ADFFE0D20",3562
 730 DATA "4178B1283DED433B442A3
944CD6E192031E523234E2346230BC5E
5CDB0162A59",2970
 740 DATA "5CCD551623225D5CD1EBC
1C5EDB0C1E10303030303F1FE4DCCE81
9ED4B3B44F1",4051
 750 DATA "180ACF0B3E10010000C3F
740ED43495C2A5D5CEB212443E52A615
C37ED52E560",3063
 760 DATA "69CD6E192006CDB819CDE
819C1793DB02828C5030303032BED5B5
35CD5CD5516",3270
 770 DATA "E122535CC1C5132A615C2
B2BEDB82A495CEBC1702B712B732B72F
1C37C41CD02",3631
 780 DATA "1AED433944DFFE0D2807F
E2C2016CD021AED433B44DFFE0D200A7
8B1C82A3944",3204
 790 DATA "ED42C8D8CF0BCD7E432A3
9447CB52807CD6E1920EF18032A535CE
52A3B447CB5",3412
 800 DATA "280ACD6E1920DDCDB8191
804ED5B4B5CE1EBCDE5192A394422495
CCD9517CFFF",3704
 810 DATA "3E02013E03F5FDCB3046C
4AF0DF1CD0116CD7E432A394422495C7
CB52805CD6E",3231
 820 DATA "1918032A535CE52A3B447
CB5280FCD6E192805ED52280AEBCDB81
91804ED5B4B",2861
 830 DATA "5CE1E5AFED52E128B5D51
10000CD6E18D7D118EE0000000000000
00000",2741
```

DISC-HILFE für Schneider CPC-464

„Dis“ ist eine sehr nützliche Hilfe beim Laden von Programmen und Disketten.
Ladet bitte das Programm in den Schneider bei angeschlossener Disc und saved es dann auf eine Diskette mit GOTO 210.
Wird das Programm nun von Disketten mit RUN“ # aufgerufen, erscheint ein Diskenteninhaltsverzeichnis, wobei das erste Programm unterstrichen ist. Der Strich kann mit den Pfeiltasten bewegt werden. Drückt man [COPY], so wird das angewählte Programm von der Disc geladen und gestartet. Drückt man anstelle von [COPY] aber [D], so wird das angewählte Programm gelöscht. (VORSICHT!!!)
Hat man auf der Diskette nicht das richtige Programm gefunden, genügt es, die Disc zu wechseln und auf [SPACE] zu drücken. Dann wird erneut ein Inhaltsverzeichnis gelesen.

```
1 REM ------------------------------------------
2 REM
3 REM       DISC-HILFE FUER SCHNEIDER CPC464
4 REM
5 REM GESCHRIEBEN VON: J.GLAUBRECHT  5024 PULHEIM 1
6 REM ------------------------------------------
7 REM
100 DATA 205,96,187,50,240,186,201,0,0
110 FOR i=47840 TO 47848:READ a:POKE i,a:NEXT:MODE 2:CAT
120 x=0:y=336
130 PLOT x,y,1:DRAWR 96,0
140 a$=INKEY$:IF a$="" THEN 140
150 a=ASC(a$):PLOT x,y,0:DRAWR 96,0
160 x=x+160*((a=242)-(A=243)):x=x+160*((x>639)-(x<0)):y=y+16*((A=241)-(A=240)):y
=y+16*((y>336)-(y<0))
170 IF a=224 OR a=100 THEN 200
180 IF a=32 THEN RUN
190 GOTO 130
200 x=INT(x/8)+1:y=INT((400-y)/16):a$="":FOR i=x TO x+12:LOCATE i,y:CALL &BAE0:a
$=a$+CHR$(PEEK(&BAF0)):NEXT:IF MID$(a$,9,1)<>"." THE
N GOTO 120 ELSE MODE 1: IF a=224 THEN RUN a$ ELSE !ERA,@a$:RUN
210 SAVE"#.bas":PRINT CHR$(7);
220 IF INKEY$="" THEN 220
230 PRINT"Abgespeichert !!"
```

CPC 464: „Druckart einstellen“

Dieses kleine Programm ermöglicht seinen Anwendern, die Druckart des Matrix Printers NLQ401 von Schneider softwaremäßig einzustellen. Damit entfällt das sonst notwendige, umständliche Betätigen des DIP-Schalters Nr. 4 am Drücker.

Somit stellt dieses kleine Programm eine willkommene Hilfe für die Anwender des CPC-464 und seiner Peripherie dar.

```
10 REM ********************************
20 REM *                              *
30 REM *     Druckart einstellen      *
40 REM *------------------------------*
50 REM *          Copyright by        *
60 REM *                              *
70 REM *       Volkhard Gruendel      *
80 REM *                              *
90 REM *           25.05.85           *
100 REM********************************
120 MODE 1
130 PRINT"****************************************"
140 PRINT"*                                      *"
150 PRINT"*          Bitte waehlen Sie:          *"
160 PRINT"*          -----------------           *"
170 PRINT"*                                      *"
180 PRINT"*          Druckart NLQ      = 1       *"
```

```
190 PRINT"*                                          *"
200 PRINT"*          NORMALDRUCK       = 2           *"
210 PRINT"*                                          *"
220 PRINT"****************************************"
230 PRINT
240 A$=CHR$(27)
250 PEN 0
260 INPUT A
270 PEN 1
280 IF A<>1 AND A<>2 THEN GOTO 430
290 IF A=1 THEN 310
300 IF A=2 THEN 370
310 PRINT #8,A$;"x";CHR$(1);
320 PRINT
330 PRINT"Drucker auf NLQ eingestellt !"
340 FOR t=1 TO 1000:NEXT
350 CLS
360 END
370 PRINT #8,A$;"x";CHR$(0);
380 PRINT
390 PRINT"Drucker auf Normaldruck eingestellt !"
400 FOR t=1 TO 1000:NEXT
410 CLS
420 END
430 CLS
440 PRINT
450 PRINT TAB(10)"FALSCHE EINGABE !"
460 FOR T=1 TO 1000 :NEXT
470 GOTO 120
```

-Spectrum: Trace und Single Step

Das hier vorgestellte Maschienprogramm stellt Trace- und ingle Step-Funktionen für das pectrum-Basic zur Verfügung. Das Programm läuft auf dem pectrum mit 16K und mit 48K AM.

s stehen vier Funktionen zur erfügung, die jeweils über die ANDOMIZE USR Adresse aufgerufen werden. Da die verwendeten Adressen nicht absolut angegeben werden können, werden in der folgenden Beschreibung zunächst symbolische Adressen benutzt. Die Berechnung der tatsächlichen Adresse wird weiter unten erläutert.

Die eigentliche Trace-Funktion (Aufruf durch RANDOMIZE USR trace) sorgt dafür, daß vor der Ausführung jeder einzelnen Basic-Zeile die Zeilennummer für etwa eine Sekunde in der linken oberen Ecke des Bildschirms angezeigt wird. Die Anzeige erfolgt immer schwarz auf weiß mit dem Standard-Spectrum-Zeichensatz, unabhängig von den gerade benutzten Farben. Auch Veränderungen des Zeichensatzes oder der Ausgaberoutine ändern daran nichts. Vor der Ausführung der jeweiligen Zeile wird der ursprüngliche Inhalt des Bildschirms wiederhergestellt. Durch Druck auf eine beliebige Taste (außer <BREAK>) läßt sich die Anzeige der Zeilennummer abkürzen – das Programm benutzt die PAUSE-Funktion. Ein Druck auf <BREAK> bewirkt den Abbruch des Basic-Programms. Auch in diesem Fall wird der ursprüngliche Bildschirminhalt wiederhergestellt.

Bei der Funktion für zeilenweisen Single Step (RANDOMIZE USR zstep) wird die Zeilennummer wie bei der Trace-Funktion angezeigt. Die Anzeige bleibt jedoch stehen, bis eine beliebige Taste außer <BREAK> gedrückt wird. Ein Druck auf <BREAK> bewirkt auch hier den Abbruch des Basic-Programms. Auch bei dieser Funktion wird der ursprüngliche Bildschirminhalt vor der Ausführung der Zeile wiederhergestellt.

Die Funktion für statementweisen Single Step (RANDOMIZE USR sstep) entspricht der für zeilenweisen Single Step. Hierbei wird jedoch vor der Ausführung jedes einzelnen Statements die Zeilen- und die Statementnummer in der Form "Zeile:Statement" angezeigt.

Die Ausschaltfunktion (RANDOMIZE USR off) schaltet die zu diesem Zeitpunkt aktivierte Funktion aus. Das Basic-Programm läuft normal weiter.

Von einer Funktion kann in die andere durch entsprechenden RANDOMIZE USR-Aufruf umgeschaltet werden, ohne vorher die Ausschaltfunktion zu benutzen.

Das Erscheinen einer Meldung (auch der OK-Meldung) sorgt für das Abschalten der aktivierten Funktion. Die Funktionen können also nur innerhalb eines Basic-Programms benutzt werden. Die Aufrufe müssen jedoch nicht innerhalb von Programmzeilen stehen. Es ist möglich, z. B. mit „RANDOMIZE USR trace: RUN" die Trace-Funktion zu aktivieren und das Basic-Programm zu starten. Auch nach einer Fehlermeldung ist z. B. mit „RANDOMIZE USR zstep: CONTINUE" die Fortführung des Programms mit wieder eingeschalteter Single Step-Funktion möglich.

Das Maschinenprogramm „trace" wird mit „LOAD ‚trace' CODE : RANDOMIZE USR 16384" in den Bildschirmspeicher geladen und gestartet. Diese beiden Befehle müssen direkt hintereinander ausgeführt werden. Durch dieses Maschinenprogramm wird RAMTOP um 458 Bytes herabgesetzt, um Platz für das Maschinenprogramm mit den Funktionen zu schaffen. Ist dies nicht möglich, so erscheint die Meldung „RAMTOP no good". Sonst aber werden alle absoluten Adressen des Maschinenprogramms umgerechnet und das Programm oberhalb des neuen RAMTOP plaziert. Bereits vorhandene Maschinenprogramme und UDGs bleiben also unversehrt erhalten. Es werden jedoch alle Variablen und der GOSUB-Stack gelöscht (wie bei CLEAR). Nachdem sich das Maschinenprogramm im Speicher befindet, wird im unteren Teil des Bildschirms die Startadresse des Programms angezeigt. Sie bleibt solange stehen, bis eine Taste gedrückt wird. Die Anzeige der Startadresse hat zwei Gründe: zum einen ist diese Startadresse ein Hinweis darauf, wo RAMTOP jetzt steht (Startadresse-1). Zum anderen können aus dieser Startadresse die Aufrufadressen der einzelnen Funktionen bestimmt werden. Es ist dabei:

trace = Startadresse + 7
zstep = Startadresse + 10
sstep = Startadresse + 13
off = Startadresse.

Das abgedruckte Basic-Programm erzeugt das Maschinenprogramm „trace". Es schreibt die Bytes aus den DATA-Zeilen in den Bildschirmspeicher und ermöglicht es, das Maschinenprogramm beliebig oft zu sichern. Das Programm meldet Fehler in den DATA-Zeilen mit Angabe der Zeilennummer. Nach der Korrektur eines solchen Fehlers ist das Programm erneut mit RUN zu starten. Sobald sich das Maschinenprogramm vollständig und korrekt im Bildschirmspeicher befindet, erscheint die bekannte Meldung „Start tape, then press any key." Das Maschinenprogramm kann nun auf Band aufgezeichnet werden. Nachdem die Aufzeichnung beendet ist, erscheint die Frage „Nochmal (j/n) ?". Wird diese mit "j" beantwortet, so erscheint wieder die Meldung "Start tape ..." und das Maschinenprogramm kann erneut aufgezeichnet werden. Wird keine weitere Aufzeichnung mehr gewünscht, so ist die Frage "Nochmal ?" mit "n" zu beantworten. Dann erscheint die OK-Meldung. Wer will, kann nun auch das Basic-Programm mit SAVE "tracegen" auf Band aufzeichnen.


```
  1 REM trace
  2 REM
  3 REM Trace und Single Step
  4 REM fuer den Spectrum
  5 REM
  6 REM Gerd Kluge
  7 REM Augustastr. 7
  8 REM 5810 Witten
  9 REM
100 REM Maschinenprogramm in
110 REM den Bildschirmspeicher
120 REM POKEn
130 LET start=16384: LET n=517
140 LET z=0: LET adr=start
150 READ z$,check: LET z=z+1
160 LET p=1: LET summe=0
170 IF LEN z$/2<>INT (LEN z$/2)
 THEN GO TO 300
180 LET hi=CODE z$(p)-48-7*(z$(
p)>="A")
190 LET lo=CODE z$(p+1)-48-7*(z
$(p+1)>="A")
200 LET byte=lo+16*hi
210 POKE adr,byte
220 LET summe=summe+byte
230 LET p=p+2: LET adr=adr+1
240 IF p<LEN z$ THEN GO TO 280
250 IF summe<>check THEN GO TO
310
260 IF adr<start+n THEN GO TO 1
50
270 GO TO 400
280 IF adr<start+n THEN GO TO 1
80
290 PRINT "Zu viele Bytes";: GO
 TO 320
300 PRINT "Falsche Anzahl Zeich
en";: GO TO 320
310 PRINT "Pruefsummenfehler";
320 PRINT " in Zeile ";10*z+490
: STOP
```

```
 370 REM
 380 REM Maschinenprogramm
 390 REM auf Band sichern
 400 SAVE "trace"CODE start,n
 410 INPUT "Nochmal (j/n) ? ";i$
 420 IF i$="" THEN GO TO 410
 430 IF i$="j" THEN GO TO 400
 440 IF i$="J" THEN GO TO 400
 470 REM
 480 REM 517 Bytes
 490 REM Maschinenprogramm
 500 DATA "ED5B4B5C2A595C2BCDE51
9CDBF162AB25C11CA01AFED52E52A655
C11320019D1",3429
 510 DATA "ED5230072AB45CED52300
2CF15ED53B25C2A3D5C4E2346EB363E2
BF9C5ED733D",3506
 520 DATA "5C2AB25C2311A340AFED5
2444DDD218F40DD6E00DD23DD6600DD2
37DB4280B5E",3399
 530 DATA "2356EB09EB722B7318E7E
D5BB25C1321A340016201EDB03EFDCD0
116ED4BB25C",3642
 540 DATA "03CD2B2DCDE32D010000C
D3D1FCFFFB340FC4016416441B741D04
1E341014169",3328
 550 DATA "410000ED7B3D5CC3761B3
E01213E02213E03320442ED7B3D5CCD5
41FD27B1BFD",2742
 560 DATA "CB0A7E201C2A425CCB7C2
009CD6E193A445C2809A7C2EC1B477EE
6C078285ACF",3274
 570 DATA "FFDFFE0D284BFE3AC28A1
CE7CDBF16FD340DFA8A1CFD7E0D3D280
53A0442FE03",3798
 580 DATA "CC6841DF0600FE0D2827F
E3A28DD4FE779FEEA281C21B940E5FEF
AC2441BCD82",4147
 590 DATA "1CDFFECBC28A1CC1EF023
8EBCDE93430BA2A555C3EC0A6C0AFFE0
1CE0056235E",4199
 600 DATA "ED53455C235E2356EB192
322555CEB225D5C571E00FD360AFF15F
D720D28AD14",3014
 610 DATA "CD8B19CAB940CF1611084
2263F0608C52E0024010A00EDB0C110F
42100580E0A",2556
 620 DATA "EDB021845C0E0EEDB02A5
15CE52A365CE521003C22365C21B415C
D1516213800",2816
 630 DATA "228D5C228F5CAF32915C4
74FCD9B0AED4B455CCD1B1A3A0442FE0
3200C3E3AD7",3013
 640 DATA "ED4B475C0600CD1B1A3E2
0D70132003A04423D28020E00CD3D1FE
122365CE1CD",2476
 650 DATA "1516210842163F0608C51
E0014010A00EDB0C110F41100580E0AE
DB011845C0E",2170
 660 DATA "0EEDB0C900",628
```

ruckt die gesamten im Programm vorhandenen Variablen aus

Variable-List für den (48K) ZX-Spectrum

Diese Utility nützt vor allem demjenigen, der lange Basic-Programme schreibt, und zwischendurch einen Blick auf die zahlreich benutzten Variablen werfen will. Man kann sie auch zur Fehlersuche einsetzen, indem man nachsieht, welche Variable einen verbotenen Wert enthält.

Nach dem Starten des Basic-Programms wird der MC-Code in den Speicher gelesen und auf Band abgespeichert. Bei einer späteren Verwendung braucht man es dann nur noch mit LOAD ""CODE 64568 laden und mit RANDOMIZE USR 64568 starten. Das MC-Programm gibt daraufhin alle Variablen aus, die durch ein Basic-Programm definiert wurden, auch die Werte werden ausgegeben. Mit POKE 64569,3 leitet man die Ausgabe auf den Drukker, mit POKE 64569,2 auf den Bildschirm.

Bei Arrays werden die Werte mit steigendem Index (von rechts nach links steigend) aufgelistet. Z. B. bei p(2,2,2) folgen die Werte für p(1,1,1), p(1,1,2) p(1,2,1), p(1,2,2), p(2,1,1), p(2,1,2), p(2,2,1), p(2,2,2).

Da sich im Programm nicht alle direkten Sprünge vermeiden ließen, muß man schon mit einem Monitor alle Sprungadressen verändern, um das Programm zu verschieben. Für das Disassemblieren muß man wissen, daß das Programm bei 64568 beginnt, 791 Bytes umfaßt, und daß die Bytes 65060 bis 65341 nur als Textspeicher dienen, und nicht als MC-Code interpretiert werden dürfen.


```
 10 REM          VARIABLE-LIST
 20 REM    BY ULRICH LAUTENSCHL
AEGER
 30 REM     DARMSTAEDTER STR. 3
8 A
 40 REM          6101 MODAUTAL 1
 50 REM          TEL. 06167/571
 70 REM
 80 REM PROGRAMM ZUM EINLESEN D
ES MC PROGRAMMS
 90 REM
100 CLEAR 64567
110 LET ADRESSE=64568
120 FOR F=1 TO 40
130 READ A$
140 FOR N=0 TO LEN A$-2 STEP 2
150 LET X=CODE A$(N+1)-48: LET
Y=CODE A$(N+2)-48
160 IF X>16 THEN LET X=X-7
170 IF Y>16 THEN LET Y=Y-7
180 LET C=16*X+Y
190 POKE ADRESSE+N/2,C
200 NEXT N
210 LET ADRESSE=ADRESSE+20
220 NEXT F
230 SAVE "V-LIST"CODE 64568,791
240 STOP
250 DATA "3E02CD01163E002100407
711014001FF17EDB02A"
260 DATA "4B5C7EFE5BDAE0FCFE7BD
A0AFDFE80C8FE9BDA30"
270 DATA "FDFEBBDA62FDFEDBDA95F
DD6803232FEE52124FE"
280 DATA "CD1CFE3E0DD7E123CDB43
3E52134FECD1CFECDE3"
290 DATA "2D3E0DD7E1CDB433E5214
6FECD1CFECDE32D3E0D"
300 DATA "D7E1CDB433E52151FECD1
CFECDE32D3E0DD72160"
310 DATA "FECD1CFEE14E2346E5CD2
B2DCDE32D3E0DD7216F"
320 DATA "FECD1CFEE1234E060023E
50BCD2B2DCDE32D3E0D"
330 DATA "D73E0DD7E1C34EFCC6203
28EFEE5217FFECD1CFE"
340 DATA "3E0DD7E1234E2346237ED
70B3E00B820F7B920F4"
350 DATA "3E0DD73E0DD723C34EFC3
2A6FE23E52191FECD1C"
360 DATA "FE3E0DD7E1CDB433E521A
8FECD1CFECDE32D3E0D"
370 DATA "D73E0DD7E1C34EFCD6203
2D4FE23E521B9FECD1C"
380 DATA "FE3E0DD7E1CDC0FDED4BF
DFEC5CDB433E5CDE32D"
390 DATA "E13E0DD7C10B3E00B1B02
0EC3E0DD7C34EFCD640"
400 DATA "F5E521FFFECD1CFEE1F1D
7237EFE7B38F9D680D7"
410 DATA "23E53E0DD72115FFCD1CF
EE1CDB433E5CDE32D3E"
420 DATA "0DD73E0DD7E1C34EFCD66
0323BFF23E52126FFCD"
430 DATA "1CFE3E0DD7E1CDC0FDED4
BFDFE7E23D73E0DD70B"
440 DATA "3E00B0B120F33E0DD7C34
EFC3E0132FDFE23234E"
450 DATA "0600E5CD2B2D21D6FECD1
CFECDE32D3E0DD721ED"
460 DATA "FECD1CFE3E28D7E146235
E2356E5D52AFDFECDA9"
470 DATA "3022FDFED1C5D5C1CD2B2
DCDE32D3E2CD7C1E110"
480 DATA "E0233A885C3C32885CE52
A845CCD3EFFE13E29D7"
490 DATA "3E0DD7C97EFE0DC8D7231
8F84C61756676617269"
500 DATA "61626C653A20000D6D6F6
D656E74616E65722057"
510 DATA "6572743A200D5A69656C7
76572743A200D536368"
520 DATA "726974747656974653A2
00D5A65696C656E6E75"
530 DATA "6D6D65723A200D4265666
5686C736E756D6D6572"
540 DATA "3A200D537472696E67766
172696162 6C65206124"
550 DATA "0D6E756D6572697363686
5205661726961626C65"
560 DATA "3A20610D616B7475656C6
C657220576572743A20"
570 DATA "0D6E756D6572697363686
5732056617269626C"
580 DATA "656E66656C643A20000D4
16E7A61686C20646572"
590 DATA "20506172616D657465723
A200D64696520506172"
600 DATA "616D657465723A200D010
06E756D657269736368"
610 DATA "652056617269626C653
A200D616B7475656C6C"
620 DATA "657220576572743A200D5
37472696E6776617269"
630 DATA "61626C656E66656C643A2
000240D7DFE0028032B"
640 DATA "18052D7CD6086722845CC
9"
```

VC-20 – „Old"-Programm

Diese Maschinenroutine rettet versehentlich mit „NEW gelöschte Basic-Programme

Eigentlich besteht das Programm „Old" aus einem 74 Byte langem Maschinensprachprogramm. Der übrige Teil des Basicprogramms sind anwenderfreundliche Arbeitsanweisungen und Speicheroperationen, die helfen das Programm im Speicher abzulegen.
Das Maschinenprogramm bewirkt, daß nach unabsichtlicher Eingabe des Befehls „New" oder nach einem Systemabsturz in Verbindung mit einem Basickaltstart (Reset) das jeweilige Programm „gerettet" wird.
Das grundlegende Prinzip ist einfach: Bei der Eingabe von „new" (oder nach einem Reset) werden alle Basiczeichen auf den Ram-Anfang gesetzt (Programmanfang), Variablenanfang, -ende usw.). Das M-Programm sucht, wenn es aufgerufen wird, nach den alten Werten und setzt alle Zeiger zurück. Wichtig ist nur, daß nach dem „Löschen" des Programms keine Variablen benutzt worden sind. Anschließend kann man mit der Korrektur (z.B.) des Programmes fortfahren.
„Old" sollte man direkt nach dem Einschalten des Computers im Speicher ablegen. Auf jeden Fall ist es empfehlenswert, es vor Überschreibung zu schützen. Dazu bietet das Programm folgende Hilfen an: Der Benutzer kann anhand von drei Menüpunkten wählen, wo er das Programm ablegen möchte. Will er es am Speicheranfang (gemeint ist der Ram-Anfang) ablegen, so drückt er den entsprechenden Menüpunkt, und das Programm setzt selbständig den Ram-Anfang hinauf und legt das M-Programm in den geschaffenen Freiraum ab.
Dann gibt das Programm den neuen Raum – Anfang auf dem Bildschirm – aus und erteilt dem Benutzer Anweisung, wie er sich nach einem Reset zu verhalten hat. (Bei einem Reset wird der Ram-Anfang auf den ursprünglichen Wert zurückgesetzt. Damit das Programm nicht überschrieben wird, muß der Anfang wieder heraufgesetzt werden). Danach meldet das Programm, in welchem Speicherbereich das M-Programm nun liegt und wie der Benutzer es aufrufen kann (mit SYS-Adresse). Dann löscht es sich selbst durch „New" (Da das Programm sich selbst überschreibt, wenn es an den Basicanfang gelegt wird, kann der Basicteil des Programms „Old" nicht zurückgeholt werden).
Möchte der Benutzer des M-Programm am Ram-Ende abspeichern, so wählt er den Buchstaben „E" des Menues und das Programm läuft ähnlich dem oben genannten ab.
Wenn der Benutzer das M-Programm außerhalb des Rams abspeichern und den Speicherbereich selber bestimmen will, so gibt er ein „?" ein. Das Programm weist ihn an, einen Bereich zu wählen, in dem das M-Programm vor Überschreibung durch einen Reset (z.B.) oder ähnliches sicher ist.
Beim VC-20 kann man eine 8-K Erweiterung mittels Schiebeschalter in den Bereich 40960 – 49151 legen (beachte die Bedienungsanleitung) und das Programm dort ablegen. Man kann es auch in den Kassettenpuffer abspeichern, es wird dann aber durch Kassettenoperationen oder durch einen Reset gelöscht.
Wenn man die Anfangsadresse dieses Bereiches eingegeben hat, so versichert sich der Computer, ob diese auch korrekt sei. Ist sie das, so wird der Maschinenspracheteil dort abgespeichert und der Computer sagt, wie man es aufruft (s.o.). Danach löscht sich das Programm selbst.
Auf jeden Fall sollte man das Programm vorher auf einen Datenträger abspeichern, sonst könnte man beim ersten Ausprobieren eine böse Überraschung erleben.
Wie schon gesagt ist das Programm auf dem VC-20 geschrieben worden, da sich aber das Maschinensprachprogramm und alle Poke und Peek auf die Zero – Page beziehen, sollte das VC-20-Programm eine Hilfe für jeden Programmierer und auch von einem absoluten Laien anwendbar sein.
P.S. Ein Basickaltstart (Reset) wird durch Verbindung des 1. Pins (Ground) mit dem 3. Pin (Reset) des User-Port erreicht. (s. Homecomputer 1/85 und Computronic 9/84).

```
 0 POKE36879,26:PRINT"■";:CLR
 1 PRINT"**********************";
 2 PRINT"*    CARSTEN BIEKER    *";
 3 PRINT"* V.BRUCHHAUSENSTR.5A*";
 4 PRINT"* 4350 RECKLINGHAUSEN*";
 5 PRINT"*  02361 / 16117      *";
 6 PRINT"**********************";
10 DATA 24, 160, 3, 200, 177, 43
20 DATA 208, 251, 200, 152, 160, 0
30 DATA 101, 43, 133, 251, 145, 43
40 DATA 152, 200, 101, 44, 133, 252
50 DATA 145, 43, 160, 0, 230, 251
60 DATA 208, 2, 230, 252, 177, 251
70 DATA 208, 244, 200, 192, 3, 208
80 DATA 247, 169, 4, 101, 251, 133
90 DATA 45, 133, 47, 133, 49, 133
100 DATA 51, 169, 0, 101, 252, 133
110 DATA 46, 133, 48, 133, 50, 133
120 DATA 52 , 169, 0, 133, 53, 133
130 DATA 54, 96
```



```
500 B=0:FORI=1TO74:READA:B=B+A:NEXT:RESTORE
510 IFB=9564THEN1000
520 PRINT"■■IN DEN DATAZEILEN IST EIN WERTEFEHLER !!!■"
530 PRINT:PRINT"■BITTE UEBERPRUEFEN SIEDIE DATAS":END
1000 PRINT:PRINT"DAS PROGRAMM ■■OLD■■ IST  74 BYTE LANG"
1010 PRINT:PRINT"WO WOLLEN SIE ES AB-  SPEICHERN  ?"
1020 PRINT:PRINT"      BASIC■■A■■NFANG"
1030 PRINT:PRINT"      BASIC■■E■■NDE"
1040 PRINT:PRINT"          ■■?■■"
1050 PRINT:PRINT:PRINT"■■ BITTE TASTE DRUECKEN ■■";
1060 GETW$:IFW$<>"A"ANDW$<>"E"ANDW$<>"?"THEN1060
1070 IFW$="A"THEN2500
1080 IFW$="E"THEN3000
2000 PRINT"■WENN SIE DEN SPEICHER-BEREICH SELBST WAEHLENWOLLEN , SOLLTEN SIE"
2010 PRINT"DAS PROGRAMM VOR UEBERSCHREIBUNG SCHUETZEN"
2020 PRINT:PRINT"WENN SIE DAS PROGRAMM IRGENDWO UNTERHALB DESBASICANFANGS AB-"
2030 PRINT"SPEICHERN (Z.B. KASSETTENPUFFER) , KANN ES  SEIN , DASS DAS PRO-"
2040 PRINT"GRAMM NACH EINEM RESETGELOESCHT IST"
2050 PRINT:PRINT:PRINT"■■ BITTE TASTE DRUECKEN ■■";
2060 GETW$:IFW$=""THEN2060
2070 PRINT"■GEBEN SIE DIE ANFANGS-ADRESSE EIN :"
2080 PRINT"■":PRINT:PRINTTAB(8)
2090 INPUTAN
2100 PRINT:PRINT"■IST DIE ANFANGSADRESSEKORREKT ? (J/N)"
2110 GETW$
2120 IFW$="N"THEN2070
2130 IFW$<>"J"THEN2110
2140 FOR I = AN TO AN+73
2150 READDA
2160 POKEI,DA
2170 NEXT
2180 GOTO4000
2500 AF=PEEK(43)+PEEK(44)*256:REM BASICANFANG
2600 PRINT"■ACHTUNG"
2610 PRINT:PRINT"UM DAS PROGRAMM VOR  UEBERSCHREIBUNG ZU  SCHUETZEN , IST DER
"
2620 PRINT"BASICANFANG HERAUF-  GESETZT WORDEN. ER  LIEGT JETZT BEI :"
2630 PRINT:PRINTTAB(8)"■■"AF+79"■■"
2640 PRINT:PRINT"WENN NACH EINEM SYSTEMABSTURZ EIN RESET  DURCHGEFUEHRT WIRD
,"
2650 PRINT"LIEGT DER BASICANFANG WIEDER BEI  ■■"AF"■■"
2660 PRINT:PRINT:PRINT"■■ BITTE TASTE DRUECKEN ■■";
2670 GETW$:IFW$=""THEN2670
2680 PRINT"■GEBEN SIE DAHER NACH  JEDEM RESET FOLGENDE  ZEILEN EIN"
2690 PRINT:PRINT"■POKE43,"AF+79"AND255:POKE44,"AF+79"/256:POKE"AF+79"-1,0:NEW■"
2700 PRINT:PRINT"UND DRUECKEN SIE RE-  TURN"
2710 PRINT:PRINT"RUFEN SIE ERST DANN  DAS MASCHINENSPRACHPROGRAMM AUF"
2720 PRINT:PRINT:PRINT"■■ BITTE TASTE DRUECKEN ■■";
2730 GETW$:IFW$=""THEN2730
2740 AN=AF+4
2750 GOTO4000
3000 EN=PEEK(55)+PEEK(56)*256:REM BASICENDE
3010 POKE55,(EN-74)AND255
3020 POKE56,(EN-74)/256
3030 CLR
3040 EN=PEEK(55)+PEEK(56)*256:REM BASICENDE
3050 FOR I =EN TO EN+73
3060 READDA
3070 POKEI,DA
3080 NEXT
```

```
3090 PRINT"■ACHTUNG"
3100 PRINT:PRINT"UM DAS PROGRAMM VOR   UEBERSCHREIBUNG ZU    SCHUETZEN , IST DAS
"
3110 PRINT"BASICENDE HERUNTER-   GESETZT WORDEN. ES    LIEGT JETZT BEI :"
3120 PRINT:PRINTTAB(8)"■■"EN"■■"
3130 PRINT:PRINT"WENN NACH EINEM SYSTEMABSTURZ EIN RESET     DURCHGEFUEHRT WIRD
,"
3140 PRINT"LIEGT DAS BASICENDE   WIEDER  BEI  ■■"EN+74"■■"
3150 PRINT:PRINT"■■ BITTE TASTE DRUECKEN ■■";
3160 GETW$:IFW$=""THEN3160
3170 PRINT"■GEBEN SIE DAHER NACH  JEDEM RESET FOLGENDE  ZEILEN EIN"
3180 PRINT:PRINT"■POKE55,"EN"AND255:POKE56,"EN"/256:CLR■"
3190 PRINT:PRINT"UND DRUECKEN SIE RE-  TURN"
3200 PRINT:PRINT"RUFEN SIE ERST DANN   DAS MASCHINENSPRACHPROGRAMM AUF"
3210 PRINT:PRINT:PRINT"■■ BITTE TASTE DRUECKEN ■■";
3220 GETW$:IFW$=""THEN3220
3230 AN=EN
4000 PRINT"■DAS PROGRAMM LIEGT IN DEM SPEICHERBEREICH   VON■"AN"■BIS■"AN+73
4010 PRINT"■(EINSCHLIESSLICH)"
4020 PRINT:PRINT"AUFGERUFEN WIRD DAS   PROGRAMM DURCH :"
4030 PRINT:PRINT:PRINT"    ■SYS"AN"■"
4040 GETW$:IFW$=""THEN4040
4050 IFAF=0THENNEW
4060 FOR I = AF+4 TO AF+4+73
4070 READDA
4080 POKEI,DA
4090 NEXT
4100 POKE43,(AF+79)AND255
4110 POKE44,(AF+79)/256
4120 POKEAF+78,0
4130 NEW
```

Atari-Programm zur Erstellung von Hardcopies auf den Epson-Druckern FX/RX-80

Der erste Schritt ist die Eingabe des Hauptprogramms. Am besten speichern Sie das fertige BASIC Programm auf den vorhandenen Datenträger. Nach dem Starten des Programms werden die Daten des Maschinenprogramms in den geschützten Bereich der 'PAGE SIX' gelesen, die bei 1536 beginnt. Ist das Programm ordnungsgemäß eingegeben worden, meldet es sich mit 'OK'. Das BASIC-Programm kann gelöscht werden, wenn die Daten richtig eingelesen wurden. Wenn nun die Grafikstufe 8+16 im Programm benutzt wird, kann durch den Befehl 'X= USR(1536)' der gesamte Bildschirminhalt auf den Drucker gebracht werden. Dazu werden nur 48 sec. benötigt. Das Demoprogramm zeigt in Kürze die Anwendung der Maschinenroutine.

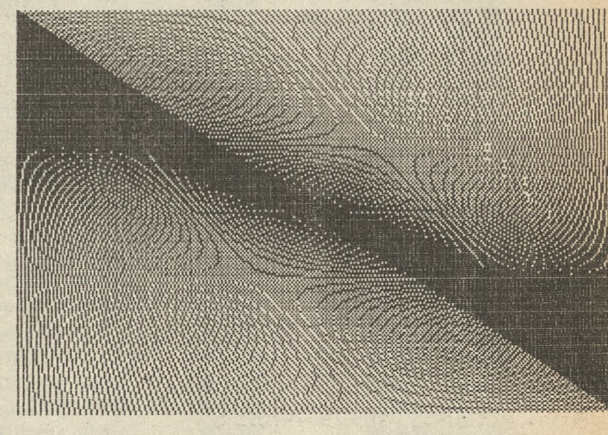
```
10 REM GRAPHIC 8+16  DRUCKERAUSGABE FUER
20 REM ATARI COMPUTER MIT FX/RX-80 DRUCKER
30 REM (C) 6.1985 BY MICHAEL SASSIN
40 REM AUF DEM KAMP 10  5163 LANGERWEHE MERODE
50 C=0
60 FOR X=1536 TO 1717
70 READ A:C=C+A:POKE X,A:NEXT X
80 IF C<>23431 THEN ? "DATENFEHLER":END
90 ? "OK":END
100 DATA 104,169,128,133,216,165,88,133,212,133,214,165,89
110 DATA 133,213,133,215,169,25,133,217,198,217,240,102,32
120 DATA 144,6,169,41,133,218,198,218,240,71,160,0,162
130 DATA 0,177,214,37,216,201,1,38,219,24,165,214,105
140 DATA 40,133,214,165,215,105,0,133,215,232,224,8,144
```



```
150 DATA 230,165,219,32,160,6,165,212,133,214,165,213,133
160 DATA 215,70,216,144,209,169,128,133,216,24,165,212,105
170 DATA 1,133,212,133,214,165,213,105,0,133,213,133,215
180 DATA 76,32,6,24,165,212,105,24,133,212,133,214,165
190 DATA 213,105,1,133,213,133,215,76,21,6,169,80,133
200 DATA 217,198,217,240,8,169,0,32,160,6,76,131,6
210 DATA 96,160,9,152,72,185,172,6,32,160,6,104,168
220 DATA 136,208,243,96,170,173,55,228,72,173,54,228,72
230 DATA 138,162,0,96,1,64,4,42,27,10,8,65,27
```

```
DEMO

10 GRAPHICS 8+16:COLOR 1
20 FOR X=0 TO 319 STEP 3
30 PLOT 0,0:DRAWTO X,191:
PLOT 319,191:DRAWTO 319-X,0
40 NEXT X
50 X=USR(1536)
```



```
                10       .OPT LIST
                20       .OPT NO EJECT
                30       .OPT OBJ
                40 ;**********************************
                50 ;* VERBESSERUNG DER GRAFIKAUSGABE *
                60 ;* FUER DEN FX/RX-80 DRUCKER      *
                70 ;* (C) 6.1984 BY MICHAEL SASSIN   *
                80 ;* LAUFZEIT : 48 SEC              *
                90 ;**********************************
    =00D4       0100 BILD =  $D4
    =00D6       0110 BILD1 = $D6
    =00D8       0120 MASKE = $D8
    =00D9       0130 ZAEH =  $D9
    =00DA       0140 ZAEH1 = $DA
    =00DB       0150 ZEICHEN = $DB
0000            0160      *=  $0600
0600 68         0170      PLA          ENTFERNE EINTRAGUNG VOM STACK
0601 A980       0180      LDA #$80     BIT-MASKE AUF MSB SETZEN
0603 85D8       0190      STA MASKE
0605 A558       0200      LDA 88       TRAGE ANFANG DES BILD-
0607 85D4       0210      STA BILD     SCHIRMES IN BILD
0609 85D6       0220      STA BILD1    UND BILD1 EIN
060B A559       0230      LDA 89
060D 85D5       0240      STA BILD+1
060F 85D7       0250      STA BILD1+1
0611 A919       0260      LDA #25      REIHENZAEHLER AUF 25 SETZEN
0613 85D9       0270      STA ZAEH
0615 C6D9       0280 ROW  DEC ZAEH     ERNIEDRIGE REIHENZAEHLER
0617 F066       0290      BEQ ENDE     SPR. WENN 0
```

```
0619 209006     0300      JSR VORSP    DRUCKERINITIATION
061C A929       0310      LDA #41      BYTE-ZAEHLER AUF 41 SETZEN
061E 85DA       0320      STA ZAEH1
0620 C6DA       0330 CHAR DEC ZAEH1    ERNIEDRIGE BYTE-ZAEHLER
0622 F047       0340      BEQ ROWEND   SPR. WENN 0
0624 A000       0350 BYTE1 LDY #0
0626 A200       0360      LDX #0       REIHENZAEHLER AUF 8 SETZEN
0628 B1D6       0370 BYTE LDA (BILD1),Y LADE 1-BIT INFORMATION
062A 25D8       0380      AND MASKE    DURCH MASKIERUNG
062C C901       0390      CMP #1       VERGLEICHE ERGEBNIS MIT 1
062E 26DB       0400      ROL ZEICHEN VERGLEICHSERGEBNIS ABSPEICH.
0630 18         0410      CLC          ERHOEHE SPEICHERZEILEN
0631 A5D6       0420      LDA BILD1    UM 40,DIE AUF DIE
0633 6928       0430      ADC #40      NAECHSTE ZEILE WEISEN
0635 85D6       0440      STA BILD1
0637 A5D7       0450      LDA BILD1+1
0639 6900       0460      ADC #0
063B 85D7       0470      STA BILD1+1
063D E8         0480      INX          ERHOEHE REIHENZAEHLER
063E E008       0490      CPX #$08     8. REIHE ERREICHT ?
0640 90E6       0500      BCC BYTE     SPR. WENN NEIN
0642 A5DB       0510      LDA ZEICHEN LADE DRUCKINFORMATION
0644 20A006     0520      JSR PRINT    DRUCKE SIE
0647 A5D4       0530      LDA BILD     STELLE ALTEN ZUSTAND
0649 85D6       0540      STA BILD1    DER SPEICHERZELLE
064B A5D5       0550      LDA BILD+1   BILD1 WIEDER HER
064D 85D7       0560      STA BILD1+1
064F 46D8       0570      LSR MASKE    UNTERSUCHE NAECHSTES BIT
0651 90D1       0580      BCC BYTE1    SPR. WENN NICHT 8 BIT UNTERSUCHT
0653 A980       0590      LDA #$80     STELLE ALTE MASKE
0655 85D8       0600      STA MASKE    WIEDER HER
0657 18         0610      CLC          ERHOEHE ZEIGER AUF
0658 A5D4       0620      LDA BILD     NAECHSTE BILDSCHIRM-
065A 6901       0630      ADC #$01     SPEICHERZELLE
065C 85D4       0640      STA BILD
065E 85D6       0650      STA BILD1
0660 A5D5       0660      LDA BILD+1
0662 6900       0670      ADC #$00
0664 85D5       0680      STA BILD+1
0666 85D7       0690      STA BILD1+1
0668 4C2006     0700      JMP CHAR     NAECHSTE ZEICHEN
066B 18         0710 ROWEND CLC        ERHOEHE DEN BILD-
066C A5D4       0720      LDA BILD     SCHIRMZEIGER UM
066E 6918       0730      ADC #24      8 ZEILEN
0670 85D4       0740      STA BILD
0672 85D6       0750      STA BILD1
0674 A5D5       0760      LDA BILD+1
0676 6901       0770      ADC #$01
0678 85D5       0780      STA BILD+1
067A 85D7       0790      STA BILD1+1
067C 4C1506     0800      JMP ROW      NAECHSE REIHE
067F A950       0810 ENDE LDA #80      DRUCKERPUFFER AUF-
0681 85D9       0820      STA ZAEH     FUELLEN
0683 C6D9       0830 ENDE1 DEC ZAEH
0685 F008       0840      BEQ END
0687 A900       0850      LDA #$00
0689 20A006     0860      JSR PRINT
068C 4C8306     0870      JMP ENDE1
068F 60         0880 END RTS
```



```
0690 A009       0890 VORSP LDY #9      ANZAHL DER STEUERZEICHEN
0692 98         0900 LOOP TYA          RETTE Y-REGISTER
0693 48         0910      PHA
0694 B9AC06     0920      LDA DAT-1,Y LADE INFORMATION
0697 20A006     0930      JSR PRINT    DRUCKE SIE
069A 68         0940      PLA          HOLE Y-REGISTER
069B A8         0950      TAY          VOM STACK
069C 88         0960      DEY          ERNIEDRIGE Y-REGISTER
069D D0F3       0970      BNE LOOP     SPR. WENN NICHT 0
069F 60         0980      RTS
06A0 AA         0990 PRINT TAX         RETTE A IN X
06A1 AD37E4     1000      LDA $E437    LADE ADRESSE FUER DIE
06A4 48         1010      PHA          1 BYTE AUSGABE DES
06A5 AD36E4     1020      LDA $E436    DRUCKERS AUF STACK
06A8 48         1030      PHA          UND SPRINGE SIE AN
06A9 8A         1040      TXA          X ZURUECK IN A
06AA A200       1050      LDX #0       KEHRE GLEICHZEITIG
06AC 60         1060      RTS          ZUM AUFRUF ZURUECK
                1070 ; DRUCKERINITIATIONSDATEN
06AD 0140042A   1080 DAT .BYTE 1,64,4,42,27,10,8,65,27
06B1 1B0A0841
06B5 1B
```

```
=00D4 BILD          =00D6 BILD1          0628 BYTE          0624 BY
TE1                 0620 CHAR
 06AD DAT           068F END             067F ENDE          0683 EN
DE1                 0692 LOOP
=00D8 MASKE         06A0 PRINT           0615 ROW           066B RO
WEND                0690 VORSP
=00D9 ZAEH          =00DA ZAEH1          =00DB ZEICHEN
```

Dieses Utility stellt Basic-Befehle für die Arbeit mit dem Bildschirm zur Verfügung

VIDEO-UTILITY für den C-64

Dieses Programm besteht aus einem Basic-Lader und einem Demo-Programm, in dem sich die Basic-Erweiterung selbst erklärt.

Gestartet wird das Programm mit dem Befehl SYS 12*4096. Als Sonderzeichen hat der Autor '!' gewählt.

Wem das Sonderzeichen nicht zusagt, der kann leicht mit POKE 49169,ASC('Zeichen') ein anderes Sonderzeichen (z. B. #,&,←) definieren.

```
1 REM************************************
2 REM*          VIDEO-UTILIES           *
3 REM*          VON JOH. KOGL           *
4 REM*       3004 RIED/RIEDERBERG       *
5 REM*          OESTERREICH             *
6 REM************************************
10 FOR I = 49152 TO 49380 :READDA:POKEI,DA:NEXTI
100 DATA 169,11,141,8,3,169,192,141,9,3,96,32,115,0,240,4,201
101 DATA 33,240,3,76,231,167,32,115,0,32,32,192,76,174,167,162,0
102 DATA 221,61,192,240,8,232,228,4,208,246,76,8,175,138,10,170,189
103 DATA 68,192,72,189,67,192,72,76,115,0,67,72,82,80,70,90,78
104 DATA 192,89,192,100,192,121,192,166,192,204,192,169,147,32,210,255,32
105 DATA 121,0,76,231,167,169,19,32,210,255,32,121,0,76,231,167,32
106 DATA 253,174,32,158,183,224,2,144,3,76,72,178,134,199,32,121,0
107 DATA 76,231,167,32,253,174,32,158,183,224,25,144,3,76,72,178,134
```

```
108 DATA 214,32,253,174,32,158,183,224,40,144,3,76,72,178,134,211,32
109 DATA 16,229,32,253,174,32,160,170,32,121,0,76,231,167,32,253,174
110 DATA 32,158,183,224,16,144,3,76,72,178,142,32,208,32,253,174,32
111 DATA 158,183,224,16,144,3,76,72,178,142,33,208,32,121,0,76,231
112 DATA 167,32,253,174,32,158,183,224,16,144,3,76,72,178,142,134,2
113 DATA 32,121,0,76,231,167,0,255
200 REM DEMO VIDEO-UTILIES
205 SYS12*4096
210 !C:!F,0,0
220 !R,1:!P,0,12,"VIDEO-UTILIES"
230 A$="VON JOH. KOGL"
240 !Z,7:!P,2,12,A$
250 !P,4,0,"DAS MC-PROGRAMM ERSETZT DIE STEUER-":!P,5,0,"ZEICHENFUMMELEI"
260 !P,6,0,"UND HAT FOLGENDEN SYNTAX:"
265 !Z,1
270 !P,8,0,"!C":!P,8,13,"= CLS (CLEAR/HOME)"
280 !P,10,0,"!H":!P,10,13,"= HOME"
290 !P,12,0,"!R,Z":!P,12,13,"= REVERSE (Z=1(AN)/0(AUS))"
300 !P,14,0,"!P,Z,S,'TEXT'":!P,14,13,"= PRINT AT (Z=ZEILE)"
310 !P,15,24,"(S=SPALTE)"
320 !P,17,0,"!F,R,H":!P,17,13,"= BS-FARBE (R=RAHMENFARBE)"
330 !P,18,24,"(H=HINTERGRFRB.)"
340 !P,20,0,"!Z,Z":!P,20,13,"= ZEICHENFARBE (Z=FARBNR.)"
350 GETA$:IFA$=""THEN350
360 !H:LIST
```

Assembler-Listing

```
    PC   IRQ  NV-BDIZC AC XR YR SP
.;  E147 EA31 10110000 93 00 00 F8
.
.,  C000  A9 0B     LDA #$0B
.,  C002  8D 08 03  STA $0308
.,  C005  A9 C0     LDA #$C0
.,  C007  8D 09 03  STA $0309
.,  C00A  60        RTS
.,  C00B  20 73 00  JSR $0073
.,  C00E  F0 04     BEQ $C014
.,  C010  C9 21     CMP #$21
.,  C012  F0 03     BEQ $C017
.,  C014  4C E7 A7  JMP $A7E7
.,  C017  20 73 00  JSR $0073
.,  C01A  20 20 C0  JSR $C020
.,  C01D  4C AE A7  JMP $A7AE
.,  C020  A2 00     LDX #$00
.,  C022  DD 3D C0  CMP $C03D,X
.,  C025  F0 08     BEQ $C02F
.,  C027  E8        INX
.,  C028  E4 04     CPX $04
.,  C02A  D0 F6     BNE $C022
.,  C02C  4C 08 AF  JMP $AF08
.,  C02F  8A        TXA
.,  C030  0A        ASL A
.,  C031  AA        TAX
.,  C032  BD 44 C0  LDA $C044,X
.,  C035  48        PHA
.,  C036  BD 43 C0  LDA $C043,X
.,  C039  48        PHA
.,  C03A  4C 73 00  JMP $0073
.,  C03D  43        ???
.,  C03E  48        PHA
.,  C03F  52        LSR???
.,  C040  50 46     BVC $C088
.,  C042  5A        LSR???
.,  C043  4E C0 59  LSR $59C0
.,  C046  C0 64     CPY #$64
.,  C048  C0 79     CPY #$79
.,  C04A  C0 A6     CPY #$A6
.,  C04C  C0 CC     CPY #$CC
.,  C04E  C0 A9     CPY #$A9
.,  C050  93        ???
.,  C051  20 D2 FF  JSR $FFD2
.,  C054  20 79 00  JSR $0079
.,  C057  4C E7 A7  JMP $A7E7
.,  C05A  A9 13     LDA #$13
.,  C05C  20 D2 FF  JSR $FFD2
.,  C05F  20 79 00  JSR $0079
.,  C062  4C E7 A7  JMP $A7E7
```



```
.,  C065  20 FD AE  JSR $AEFD
.,  C068  20 9E B7  JSR $B79E
.,  C06B  E0 02     CPX #$02
.,  C06D  90 03     BCC $C072
.,  C06F  4C 48 B2  JMP $B248
.,  C072  86 C7     STX $C7
.,  C074  20 79 00  JSR $0079
.,  C077  4C E7 A7  JMP $A7E7
.,  C07A  20 FD AE  JSR $AEFD
.,  C07D  20 9E B7  JSR $B79E
.,  C080  E0 19     CPX #$19
.,  C082  90 03     BCC $C087
.,  C084  4C 48 B2  JMP $B248
.,  C087  86 D6     STX $D6
.,  C089  20 FD AE  JSR $AEFD
.,  C08C  20 9E B7  JSR $B79E
.,  C08F  E0 28     CPX #$28
.,  C091  90 03     BCC $C096
.,  C093  4C 48 B2  JMP $B248
.,  C096  86 D3     STX $D3
.,  C098  20 10 E5  JSR $E510
.,  C09B  20 FD AE  JSR $AEFD
.,  C09E  20 A0 AA  JSR $AAA0
.,  C0A1  20 79 00  JSR $0079
.,  C0A4  4C E7 A7  JMP $A7E7
.,  C0A7  20 FD AE  JSR $AEFD
.,  C0AA  20 9E B7  JSR $B79E
.,  C0AD  E0 10     CPX #$10
.,  C0AF  90 03     BCC $C0B4
.,  C0B1  4C 48 B2  JMP $B248
.,  C0B4  8E 20 D0  STX $D020
.,  C0B7  20 FD AE  JSR $AEFD
.,  C0BA  20 9E B7  JSR $B79E
.,  C0BD  E0 10     CPX #$10
.,  C0BF  90 03     BCC $C0C4
.,  C0C1  4C 48 B2  JMP $B248
.,  C0C4  8E 21 D0  STX $D021
.,  C0C7  20 79 00  JSR $0079
.,  C0CA  4C E7 A7  JMP $A7E7
.,  C0CD  20 FD AE  JSR $AEFD
.,  C0D0  20 9E B7  JSR $B79E
.,  C0D3  E0 10     CPX #$10
.,  C0D5  90 03     BCC $C0DA
.,  C0D7  4C 48 B2  JMP $B248
.,  C0DA  8E 86 02  STX $0286
.,  C0DD  20 79 00  JSR $0079
.,  C0E0  4C E7 A7  JMP $A7E7
.,  C0E3  00        BRK
.,  C0E4  FF        ???
.
.
```

Basic-RWTS-Befehle für den Apple II

Maschinenprogramme können beim Apple II mit Hilfe der RWTS-Routine auf einzelne Sektoren der Disketten zugreifen. Mit Hilfe des hier abgedruckten Maschinenprogramms ist es nun auch BASIC-Programmierern möglich, die vielseitigen Funktionen der RWTS-Routine zu nutzen.
BASIC-RWTS erlaubt den Zugriff auf beliebige Sektoren von BASIC-Programmen. Dazu wird mit Hilfe der Ampersand-Funktion (&) der Befehlssatz des Interpreters erweitert.

Die Möglichkeiten der BASIC-RWTS-Maschinenroutine:
BASIC-RWTS bietet folgende Befehle:
&READ <Buffer>,<Track>,<Sektor>
&READ liest einen Sektor der Diskette in einen frei wählbaren Speicherbereich (Buffer) ein.
Als Parameter müssen die Anfangsadresse des 256-Byte-Buffers, der Track und der Sektor angegeben werden.
Diese Werte müssen im Dezimalsystem an den &READ-Befehl angefügt werden.
Die Spurnummer muß im Bereich zwischen 0 und 40 (für 40-Track-Laufwerke) liegen, ansonsten wird eine Fehlermeldung ausgegeben.
Beispiel:
&READ 16384 , 17 , 15
Sektor 15 auf Spur 17 (der erste CATALOG-Sektor) wird in den Speicherbereich von 16384 bis 16639 ($4000-$40FF) eingelesen.
&STORE <Buffer>,<Track>,<Sektor>
Der Inhalt des 256-Byte-Buffers wird auf der Diskette in dem angegebenen Sektor des Track gespeichert.
Beispiel:
&STORE 32768 , 17 , 0
Auf Sektor 0 der Spur 17 (dem VTOC-Sektor) wird der Inhalt des Speicherbereichs von 32768 bis 33023 ($8000 bis $80FF) abgelegt.
&S <Slot>
ändert den Slot, den &READ und &STORE verwenden.
Beispiel: &S 6 schaltet auf Slot 6 um.
&D <Drive>
ändert die Nummer des Diskettenlaufwerks, welches &READ und &STORE ansprechen.

Beispiel: &D 2 schaltet auf Drive 2 um.
&V <Volume>
ändert die Volume-Nummer, die bei der zu verarbeitenden Diskette erwartet wird. &V 0 läßt beliebige Disketten-Volume-Nummern zu.
Beispiel: &V 254
Bei allen Funktionen können die Parameter sowohl aus Zahlenkonstanten als auch aus Variablen und mathematischen Ausdrükken bestehen.
Tritt bei den Befehlen &READ oder &STORE ein Fehler auf, so wird eine Fehlerkennummer in der Speicherstelle 222 abgespeichert. Deshalb sollte nach jedem Aufruf dieser Routinen diese Adresse mit PEEK (222) abgefragt werden.

Folgende Fehlernummern können in Adresse 222 abgelegt werden:

PEEK (222) = 0 : Kein Fehler
= 16 : Diskette ist schreibgeschützt
= 32 : Falsche Volume-Nummer
= 64 : I/0 Error

Die Maschinenroutine BASIC-RWTS setzt einenApple II mit 48 KByte RAM und DOS 3.3 voraus.
Die Eingabe des Programms auch für Nur-Basic-Programmierer ist einfach, da das Programm mit einem Data-Generator in die Data-Zeilen gebracht wurde und nun als Basic-Programm eingetippt werden kann. Das Basic-Programm speichert das Maschinenprogramm auch gleich mit der richtigen Länge ab und überprüft natürlich vorher die richtige, fehlerfreie Eingabe.
Nach dem Start der Maschinenroutine mit RUN BASIC-RWTS schützt sich das Programm selbst gegen überschreiben, indem es HIMEM herabsetzt.

```
10  REM ************************
    *********
20  REM *
              *
30  REM *      B A S I C - R W T S
              *
40  REM *        Copyright 1985 by
              *
50  REM *        Oliver Steinmeier
              *
60  REM *          Mauerstra^e 1
              *
70  REM *        4934 Horn-B. Mbg.
    1         *
80  REM *
              *
90  REM ************************
    *********
100 TEXT : HOME : INVERSE : PRINT
    "BASIC-RWTS (C) 1985 BY OLIV
    ER STEINMEIER"
105 NORMAL
110 VTAB 10: PRINT  TAB( 14);"BI
    TTE WARTEN"
115 PRINT : PRINT
120 FOR I = 36608 TO 36829
130 READ X: POKE I,X:S = S + X
140 NEXT I
150 IF S <  > 32177 THEN  PRINT
     CHR$ (7);"FEHLER IM DATA-TE
    IL !": END
160 PRINT : PRINT "ALLES OKAY -
    KEINE FEHLER IM DATA-TEIL !"

170 PRINT : INPUT "FILE-NAME ZUM
     ABSPEICHERN AUF DISKETTE: "
    ;F$
180 PRINT : PRINT  CHR$ (4);"BSA
    VE";F$;",A$8F00,L$DE"
190 PRINT "VIEL SPA~ MIT DER BAS
    IC-ERWEITERUNG     B A S I C
       R W T S  ! ! !"
210 PRINT "**********************
    ***********"
220 PRINT "* NEUE BEFEHLE DER ER
    WEITERUNG *"
230 PRINT "* BASIC-RWTS FUER DEN
     APPLE II *"
240 PRINT "* &V<VALUE> SETZT VOL
    UME-NO.    *"
250 PRINT "* &S<VALUE> SETZT SLO
    T-NO.      *"
260 PRINT "* &D<VALUE> SETZT DRI
    VE-NO.     *"
270 PRINT "* &READ<BUFFER>,<TRAC
    K>,<SEC>   *"
280 PRINT "* LIEST SEKTOR DES TR
    ACKS IN    *"
290 PRINT "* DEN ANGEGEBENEN BUF
    FER        *"
300 PRINT "* &STORE<BUFFER>,<TRA
    CK>,<SEC> *"
310 PRINT "* SCHREIBT SEKTOR AUS
     BUFFER    *"
320 PRINT "* AUF DEN TRACK UND S
    EKTOR      *"
330 PRINT "* SPEICHERSTELLE 222
    ENTHAELT   *"
340 PRINT "* DEN FEHLERCODE (O N
    O ERROR)   *"
350 PRINT "**********************
    ***********"
360 PRINT : CALL 36608
60001  DATA 169,143,141,247,03,13
    3,116,169,29,141,246,03,133,
    115,169,76,141,245,03,169,18
    1,160,143,32,58,219,76,03,22
```



```
    4,201,135,240,19,201,168,240
    ,19,201,83,240,79,201,68,240
    ,101,201,86,240,120,76
60002  DATA 201,222,169,01,208,02
    ,169,02,141,244,183,32,177,0
    0,32,70,231,165,80,141,240,1
    83,165,81,141,241,183,224,41
    ,144,03,76,153,225,142,236,1
    83,32,76,231,224,16,144,03,7
    6,153,225,142,237,183
60003  DATA 32,227,03,32,217,03,1
    76,04,169,00,240,03,173,245,
    183,133,222,76,149,217,32,17
    7,00,32,248,230,224,08,144,0
    3,76,153,225,160,04,138,10,1
    36,208,252,141,233,183,76,14
    9,217,32,177,00,32
60004  DATA 248,230,224,01,240,07
    ,224,02,240,03,76,153,225,14
    2,234,183,76,149,217,32,177,
    00,32,248,230,142,235,183,76
    ,149,217,141,141,194,193,211
    ,201,195,173,210,215,212,211
    ,173,195,207,205,205,193,206

60005  DATA 196,211,160,168,195,1
    69,160,207,174,160,211,212,1
    97,201,206,205,197,201,197,2
    10,141,00
```

Fortsetzung des Programms „Digital Sound Editor“

```
D-S-E HEXDUMPS
HEXDUMP NR.1
ÜCALL-151
*4000.4147

4000- A5 FD 85 FC A9 00 85 FB
4008- A2 00 A9 00 81 FB 18 A5
4010- FB 69 01 85 FB A5 FC 69
4018- 00 85 FC C5 FE D0 EB A5
4020- FD 85 FC 20 DD FB A2 05
4028- AD 60 C0 29 80 85 FA AD
4030- 60 C0 29 80 C5 FA 85 FA
4038- F0 F5 CA D0 F2 A9 01 85
4040- F9 A5 FD 85 FC AD 60 C0
4048- 30 05 A9 00 4C 54 40 A5
4050- F9 4C 54 40 01 FB 81 FB
4058- 20 7E 40 18 A5 FB 69 01
4060- 85 FB A5 FC 69 00 85 FC
4068- C5 FE D0 D9 A5 F9 C9 80
4070- F0 06 0A 85 F9 4C 41 40
4078- 20 DD FB 60 69 FF A6 FF
4080- CA D0 FD 60 46 A5 8D BC
4088- B5 20 A8 A6 20 EA A2 4C
4090- 7D A2 A0 13 B1 42 D0 14
4098- C8 C0 17 D0 F7 A0 19 B1
40A0- 42 99 A4 B5 C8 C0 1D D0
40A8- F6 4C BC A6 A2 FF 8E 5D
40B0- B6 D0 F6 00 00 00 00 00
40B8- 00 00 00 00 00 00 00 00
40C0- 00 00 00 00 00 00 00 00
40C8- 00 00 00 00 00 00 00 00
40D0- 20 58 FC A9 C2 20 ED FD
40D8- A9 01 20 DA FD A9 AD 20
40E0- ED FD A9 00 20 DA FD 60
40E8- 00 00 00 00 00 00 00 00
40F0- 00 00 00 00 00 00 00 00
40F8- 00 00 00 00 00 00 B7 FF
4100- A9 01 85 F9 A2 00 A9 00
4108- 85 FB A5 FD 85 FC A1 FB
4110- 25 F9 C5 FA 85 FA F0 06
4118- 2C 30 C0 4C 23 41 EA EA
4120- 4C 23 41 20 7E 40 18 A5
4128- FB 69 01 85 FB A5 FC 69
4130- 00 85 FC C5 FE D0 D7 A5
4138- F9 C9 80 F0 06 0A 85 F9
4140- 4C 0A 41 60 69 FF 00 00
```

```
HEXDUMP NR.2
Ü
ÜCALL-151
*4200.489F

4200- 3B 00 90 00 9A 00 AE 00
4208- C3 00 D0 00 E4 00 F8 00
4210- 0C 01 20 01 34 01 48 01
4218- 5C 01 70 01 84 01 98 01
4220- AC 01 C0 01 D4 01 E8 01
4228- FC 01 10 02 24 02 38 02
4230- 4C 02 60 02 74 02 88 02
```

```
4238- 9C 02 B0 02 C4 02 D8 02
4240- EC 02 00 03 14 03 28 03
4248- 3C 03 50 03 64 03 78 03
4250- 8C 03 A0 03 B4 03 C8 03
4258- DC 03 F0 03 04 04 18 04
4260- 2C 04 40 04 54 04 68 04
4268- 7C 04 90 04 A4 04 B8 04
4270- CC 04 E0 04 F4 04 08 05
4278- 00 00 00 00 00 00 00 00
4280- 00 00 00 00 00 00 00 00
4288- 00 00 00 00 00 00 00 00
4290- 09 09 09 01 00 00 00 00
4298- 00 00 49 04 20 24 34 12
42A0- 12 0A 49 00 00 00 00 00
42A8- 00 00 00 00 00 00 48 08
42B0- 18 20 34 0A 21 34 12 12
42B8- 0A 09 00 00 00 00 00 00
42C0- 00 00 00 5D 28 28 1A 1A
42C8- 2D 2D 15 1B 1B 33 31 00
42D0- 08 2D 2D 20 3B 3F 20 29
42D8- 2D 1F 30 36 36 26 4A 09
42E0- 00 00 00 00 08 0C 0C 0C
42E8- 0C DC 3B 2E 94 92 09 35
42F0- 27 4A 01 00 00 00 00 00
42F8- 08 64 0C 24 3B 32 0E 0A
4300- 0E 0C 16 1F 2F 09 09 00
4308- 00 00 00 00 48 08 08 18
4310- 24 16 12 12 09 09 00 00
4318- 00 00 00 00 00 00 00 00
4320- 49 1C 1C 24 0C 0C 14 12
4328- 12 12 09 09 00 00 00 00
4330- 00 00 00 00 49 0C 0C 24
4338- 1C 1C 14 12 12 12 09 09
4340- 00 00 00 00 00 00 00 00
4348- 08 0C 04 E0 4C 36 36 36
4350- 0C 21 23 20 21 12 12 12
4358- 09 00 00 00 49 20 24 34
4360- 1A 2B 2D 3D 12 0A 09 00
4368- 00 00 00 00 00 00 00 00
4370- 09 0C 34 4A 09 00 00 00
4378- 00 00 00 00 00 00 00 00
4380- 00 00 00 00 48 18 28 2D
4388- 3D 12 0A 09 00 00 00 00
4390- 00 00 00 00 00 00 00 00
4398- 49 14 09 09 00 00 00 00
43A0- 00 00 00 00 00 00 00 00
43A8- 00 00 00 00 08 0C 0C 0C
```

```
43B0- 0C 94 12 12 09 00 00 00
43B8- 00 00 00 00 00 00 00 00
43C0- 08 24 24 0C 2D 15 36 36
43C8- 1E 3F 47 20 21 21 12 12
43D0- 0A 09 00 00 09 2D 27 24
43D8- 24 F4 15 12 12 09 09 00
43E0- 00 00 00 00 00 00 00 00
43E8- 21 0C 0C 65 24 3B 3F 32
43F0- 12 12 2D 2D 0F 09 00 00
43F8- 00 00 00 00 09 2D 05 20
4400- 1C 2F 20 21 3F 3F 9D 92
4408- 32 09 09 09 00 00 00 00
4410- 49 21 24 24 34 33 33 33
4418- 2D 2D 17 0A 09 00 00 00
4420- 00 00 00 00 08 0E 2D 05
4428- 20 24 3B 3F 24 2D 2D 97
4430- 92 0A 09 00 00 00 00 00
4438- 08 24 2D AD 36 3B 3F 20
4440- 24 0C 0C 2D 17 12 12 0A
4448- 09 00 00 00 09 24 0C 0C
4450- 0C 3C 3F 2F 92 12 0A 09
4458- 09 00 00 00 00 00 00 00
4460- 08 24 20 0C 2D 15 36 32
4468- 1E 3F 07 20 0C 2D 15 12
4470- 09 00 00 00 29 2D 20 21
4478- 24 1C 3F 17 36 29 2D 12
4480- 4A 00 00 00 00 00 00 00
4488- 40 40 B1 56 49 00 00 00
4490- 00 00 00 00 00 00 00 00
4498- 00 00 00 00 09 0C C4 08
44A0- 16 56 49 00 00 00 00 00
44A8- 00 00 00 00 00 00 00 00
44B0- 09 09 1C 1C 1C 0C 0C 0C
44B8- 94 92 52 09 00 00 00 00
44C0- 00 00 00 00 40 2D 2D C7
44C8- D8 2D 2D 97 52 09 00 00
44D0- 00 00 00 00 00 00 00 00
44D8- 09 0C 0C 0C 1C 1C 1C 94
44E0- 92 52 49 01 00 00 00 00
44E8- 00 00 00 00 49 04 20 0C
44F0- 0C 1C DB 62 2D 95 92 4A
44F8- 00 00 00 00 00 00 00 00
4500- 49 09 3F 3F 20 24 64 2D
4508- 15 36 37 27 2C 92 4A 01
4510- 00 00 00 00 21 24 34 08
4518- 30 40 0E 0E 36 36 C4 DB
4520- 2D AD 4A 00 00 00 00 00
```



```
4528- 29 2D DF 24 2C 2D DF 24
4530- 2C 2D 4F 32 16 36 09 00
4538- 00 00 00 00 09 07 30 20
4540- 24 24 79 2D 15 94 92 1E
4548- 1B 2D 4D 00 00 00 00 00
4550- 49 39 3F 24 24 24 2D 7D
4558- 11 36 36 94 09 00 00 00
4560- 00 00 00 00 21 24 24 2C
4568- 2D BD 12 3F 97 2A 2D 7D
4570- 09 00 00 00 00 00 00 00
4578- 21 24 24 2C 2D BD 12 3F
4580- 97 4A 49 01 00 00 00 00
4588- 00 00 00 00 08 06 20 24
4590- 64 0F 2D BD 92 0F 35 3E
4598- 3F 4F 49 01 00 00 00 00
45A0- 21 24 24 B4 2A 2D 25 24
45A8- 96 36 26 4A 00 00 00 00
45B0- 00 00 00 00 09 25 24 24
45B8- 3C 0D 97 92 0A 4F 09 00
45C0- 00 00 00 00 00 00 00 00
45C8- 08 0E 2D 4F 20 24 24 96
45D0- 12 4E 00 00 00 00 00 00
45D8- 00 00 00 00 21 24 24 74
45E0- 48 31 33 33 33 A1 0A 94
45E8- A1 09 00 00 00 00 00 00
45F0- 21 24 24 B4 92 2A 2D 3D
45F8- 49 00 00 00 00 00 00 00
4600- 00 00 00 00 21 24 24 74
4608- 0E 66 30 40 36 36 36 54
4610- 01 00 00 00 00 00 00 00
4618- 21 24 24 B4 A1 0A 94 61
4620- 24 34 12 36 A6 09 00 00
4628- 00 00 00 00 08 24 24 0C
4630- 2D 15 36 36 1E 3F 4F 49
4638- 01 00 00 00 00 00 00 00
4640- 21 24 24 2C 2D 4F A2 F6
4648- 3F 97 4A 49 01 00 00 00
4650- 00 00 00 00 08 24 24 0C
4658- 2D 15 36 A6 12 1C 1C 94
4660- 3A 4F 49 01 00 00 00 00
4668- 21 24 24 2C 2D 4F A2 F6
4670- 3F 4F A2 0A 94 A1 09 00
4678- 00 00 00 00 08 0E 2D 05
4680- 20 1C 3F 07 30 20 0C 2D
4688- 15 94 92 4E 00 00 00 00
4690- 49 24 24 24 3F 0D 2D 97
4698- 92 4A 01 00 00 00 00 00
```

```
46A0- 00 00 00 00 08 06 20 24
46A8- 24 46 49 36 36 F6 3F 4D
46B0- 09 00 00 00 00 00 00 00
46B8- 40 24 24 96 52 94 61 0C
46C0- 24 24 96 92 09 00 00 00
46C8- 00 00 00 00 21 24 24 B4
46D0- 92 61 B4 61 24 24 96 32
46D8- A6 09 00 00 00 00 00 00
46E0- 08 66 30 40 0C 44 E6 DB
46E8- A6 0A 94 71 0E A6 09 00
46F0- 00 00 00 00 49 24 24 E3
46F8- B4 49 0C B4 92 4A 00 00
4700- 00 00 00 00 00 00 00 00
4708- 08 0C 0C 0C 0C 3C 3F EF
4710- 92 12 2E 2D 7D 09 00 2D
4718- 7D 09 00 09 00 00 00 54
4720- 41 53 54 45 20 2A 2A 00
4728- 3C 0D B6 0D AD 49 24 D0
4730- 22 50 22 C4 8C 34 33 35
4738- 32 3A B1 00 5D 0D C0 0D
4740- AD 49 24 D0 22 2B 22 CE
4748- 49 24 D0 22 3B 22 C4 91
4750- 3A 50 D0 50 C8 31 3A AB
4758- 33 30 30 30 00 7B 0D CA
4760- 0D AD 49 24 D0 22 2D 22
4768- CD 50 CF 30 C4 91 3A 50
4770- D0 50 C9 31 3A AB 33 30
4778- 30 30 00 91 0D D4 0D AD
4780- 49 24 D0 22 43 22 C4 90
4788- 3A 91 3A AB 33 30 30 30
4790- 00 D6 0D DE 0D AD 49 24
4798- D0 22 47 22 C4 89 3A 97
47A0- 3A 84 22 47 45 53 43 48
47A8- 57 49 4E 44 49 47 4B 45
47B0- 49 54 3A 22 3B 47 45 3A
47B8- B9 32 35 35 2C 47 45 3A
47C0- B9 C9 31 36 33 30 34 2C
47C8- 30 3A B9 C9 31 36 32 39
47D0- 37 2C 30 3A B1 00 EF 0D
47D8- E8 0D AD 49 24 D0 22 2E
47E0- 22 CD 5A 46 D1 34 C4 5A
47E8- 46 D0 5A 46 C8 31 00 08
47F0- 0E F2 0D AD 49 24 D0 22
47F8- 2C 22 CD 5A 47 45 53 43
4800- 48 57 49 4E 44 49 47 4B
4808- 45 49 54 20 28 31 3D 53
4810- 43 48 4E 45 4C 4C 20 31
```

```
4818- 30 3D 4C 41 4E 47 53 41
4820- 4D 29 3A 22 3B 47 00 3A
4828- 0E D6 0B AD 47 D1 31 CE
4830- 47 CF 31 30 C4 33 30 32
4838- 30 00 45 0E E0 0B B9 32
4840- 35 35 2C 47 00 4D 0E EA
4848- 0B AB 31 35 00 53 0E A0
4850- 0F B2 00 59 0E B4 0F 97
4858- 00 7C 0E BE 0F 84 22 41
```

```
4860- 42 53 50 45 49 43 48 45
4868- 52 4E 20 2D 20 46 49 4C
4870- 45 4E 41 4D 45 3A 22 3B
4878- 46 49 24 00 8D 0E C8 0F
4880- AD 46 49 24 D0 22 22 C4
4888- 34 30 32 30 00 B4 0E D2
4890- 0F BA E7 28 34 29 3B 22
4898- 42 53 41 56 45 22 3B 46
```

Eingabemodul für den TI-99/4A mit oder ohne X-Basic

Diese Schaltung ermöglicht das Betreiben eines I/O-Ports auch ohne 32K-Erweiterung und ohne X-Basic

Eine solche 8-Bit-Eingabe ist nur über den Joystickport möglich, und zwar in der Form 2x4 Bit. Es können auch 1x6 Bit oder andere Kombinationen gewählt werden, je nach Einsatzgebiet.

Die beiden Eingänge für die Feuerknöpfe legen die beiden 4-Bit-Kombinationen fest, die angesprochen werden sollen. Dabei muß man aber sicherstellen, daß nur ein Eingang geschaltet ist, da sich sonst die Feuerknopfeingänge gegenseitig sperren und somit auch kein X/Y-Signal durchdingt.

Zum Schaltplan:

Die beiden Datenbits D0 und D1 werden also zur Ansteuerung der beiden Bitblöcke $\overline{D2}$-$\overline{D5}$ und D2-D5 benutzt.

Wenn D0 angesteuert ist, dann liest der Rechner den Datenblock D2-D5. Wenn D1 angesteuert ist, dann liest der den Datenblock $\overline{D2}$-$\overline{D5}$.

Zur linken Seite der Schaltung ist nur soviel zu sagen, daß dort die Kontakte sitzen, die die Kanäle schalten. Wenn man Eingänge mit niedrigen Eingangsspannungen hat, muß über eine kleine Verstärkerschaltung das entsprechende Relais angesteuert werden.

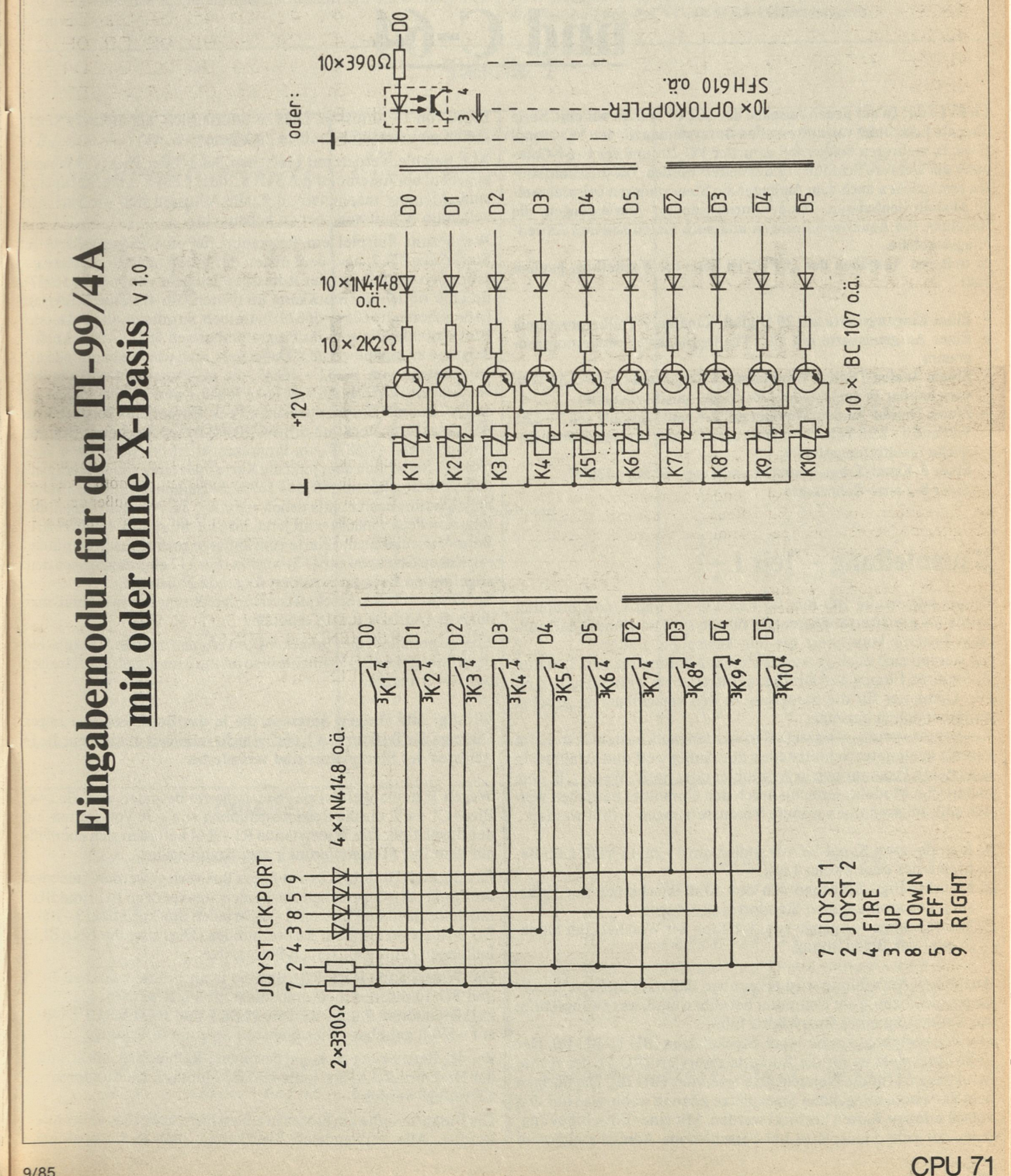
Wenn man aber z. B. wissen will, wo im Haus das Licht brennt, muß man direkt an den zu prüfenden Verbraucher ein 220-V-Relais klemmen und die Schaltkontakte, wie auf der linken Seite der Schaltung, anklemmen. Wem Relais zu laut sind, der kann natürlich auch Optokoppler als Schaltelemete nehmen oder einen Transistor benutzen.

Die gesamten Teile, die man für die Schaltung benötigt, sind bereits eingebaut, außer der 9-poligen Trapezbuchse für den Joystickport.

Die Software zur Ansteuerung:

```
100 REM ABFRAGE OB DO ODER D1 GESETZT SIND MIT
    SPRUNG IN DIE ENTSPRECHENDE ROUTINE
110 CALL KEY(1,K1,S1):: CALL KEY(2,K2,S2)::
    IF K1=18 THEN GOSUB 1000 ELSE IF K2 =18
    GOSUB 2000
120 GOTO 110
999 REM ROUTINE ZUM EINLESEN DER WERTE D2-D5
1000 CALL JOYST(1.X1,Y1):: CALL JOYST(2,X2,Y2)::
    IF X1=4 THEN D2=1 ELSE D2=0
1010 IF X1=4 THEN D3=1 ELSE D3=0
1020 IF Y2=4 THEN D4=1 ELSE D4=0
1030 IF X2=4 THEN D5=1 ELSE D5=0
1040 REM DIE DATEN LIEGEN JETZT IN D2-D5 ZUR
    VERFÜGUNG
1050 RETURN
1999 REM ROUTINE ZUM EINLESEN DER WERTE D2-D5
2000 CALL JOYST(1,X1,Y1):: CALL JOYST(2,X2,Y2)::
    IF Y1=-4 THEN DD2=1 ELSE DD2=0
2010 IF X1=-4 THEN DD3=1 ELSE DD3=0
2020 IF X2=-4 THEN DD4=1 ELSE DD4=0
2030 IF X2=-4 THEN DD5=1 ELSE DD5=0
2040 REM DIE DATEN LIEGEN JETZT IN DD2-DD5 ZUR
    VERFÜGUNG
2050 RETURN
```

Das Programm läßt sich leicht auf TI-Basic umschreiben und natürlich auch erweitern. Der Phantasie sind hier keine Grenzen gesetzt. Wem diese Eingabemethode dennoch zu kompliziert ist, der nimmt einfach nur die Kanäle D1-D5 und hat so reale 5 Bit paralell.



Universelles Bussystem für VC-20 und C-64

Liebe Leser! In der neuen Ausgabe der „CPU" starten wir eine Serie über die Erstellung von universellen Bussystemen für den VC-20 und C-64. In mehreren Folgen soll dann der VC-20- und der C-64-Computer um mehrere Schnittstellen erweitert werden. Diese Schnittstellen ermöglichen auch dem Anfänger die Steuerung von beispielsweise Modelleisenbahnen, Temperaturmessungen, Alarmanlagen, die Regelung von Aquarienheizungen und noch vielen anderen Anwendungsbereiche.
In weiteren Ausgaben der „CPU" ist dann die Vorstellung geplant von:

- **Einer Eingangskarte mit 20 Digital-Kanälen + Treiberprogramm**
- **Einer Ausgangskarte mit 18 TTL-Ausgängen + Steuerungsprogramm**
- **Einem Analog-Digital/Wandler mit 8 Eingängen von 0 – 2,55 Volt, umschaltbar in verschiedene Bereiche (positiv/negativ)**
- **Einem Digital-Analog/Wandler mit 2-Kanal-Ausgang von 0 – 2,55 Volt und 0 – 25,5 Volt / 1 Ampere, umschaltbar in verschiedene Bereiche (positiv/negativ)**
- **Einer 8-Kanal-Temperatursteuerung von -50 bis +120°C**
- **Einer 8-Kanal-Relaiskarte**

Bauanleitung – Teil 1 –

Dieser universelle I/O-Bus für den VC-20- und C-64-Computer wurde als preiswerte Alternative für die sich im Handel befindlichen Systeme konzepiert, die sehr kostspielig sind.
Sie werden sich nach geraumer Zeit der Benutzung von Tastatur, Bildschirm, Floppy und Drucker gefragt haben, wie man nun so einen Computer für die Steuerung in den verschiedenen Anwendungsbereichen einsetzt.
Die Voraussetzung, die geschaffen werden muß, besteht erstens im Einsatz geeigneter Schnittstellen zur Außenwelt und zweitens in geeigneten Programmen in Assembler oder Basic. Wenn z. B. eine elektrische Modelleisenbahn durch den Computer betrieben werden soll, müssen noch weitere Voraussetzungen erfüllt werden.

1. Ein oder zwei Kanal-D/A-Wandler von 0 – ca. 16 VDC zum Betrieb einer oder zweier Loks.
2. Mehrkanalige Eingänge von den Kontaktschienen zur Erfassung des momentanen Standortes des Zuges.
3. Mehrkanalige Ausgänge zum Schalten der Weichen, der Signale oder der Beleuchtung.

Aus dieser Aufzählung wird erkennbar, daß man mehrere Steckkarten gleichzeitig am Computer betreiben muß, was zwangsläufig zur Benutzung eines Bussystems führt.
Der Bus besteht aus zehn 31pol. Steckplätzen (BU 1 – BU 10). Der linke Steckplatz ist für die Buskarte reserviert (BU 1), der rechte Steckplatz ist für die Netzteilkarte reserviert (BU 10). Die übrigen acht dazwischenliegenden Steckplätze können wahlweise mit den verschiedenen Karten bestückt werden. Mit einer Einschränkung; jeder der acht Steckplätze hat verschiedene Adressbereiche im Block von 16 Adressen. Dies ist erforderlich, um mehrere Steckkarten gleichzeitig betreiben zu können.

Alle Befehle werden mit einfachen Poke- oder Peek(x)-Adressen gegeben; bei Assembler mit STA $...oder LDA $..., die Steckplätze sind „Memory mappened", d. h. alle Adressen sind wie Speicherzellen im Arbeitsspeicher zu behandeln.

Wenn zum Beispiel ein Programm für eine Eingangskarte im Steckplatz BU 2, also dem ersten der freien acht Steckplätze geschrieben worden ist (der Adressbereich geht von 56832 – 56847), und Sie wollen die Steckkarte im dritten (Bu 4) freien Steckplatz (Adressbereich 56864 – 56879) betreiben, so müssen alle Poke-oder Peek-Befehle, die sich im Programmbereich 56832 – 56847 befinden, neu angepaßt werden. Dabei geht man wie folgt vor: Jede der 16 Adressen von 56832 – 56847, die im Programm vorkommen, werden mit 2x16 erhöht, z. B. Poke 56840,1 wird in Poke 56872,1 geändert. Das gleiche gilt ebenso für Peek. Ein anderes Beispiel: Eine Ausgangskarte steckt im dritten (BU 4) freien Steckplatz (Adressbereich 56864 – 56879), im Programm steht der Befehl A = Peek (56864), Sie wollen aber jetzt die Karte im letzten freien Steckplatz (BU 9), Adressbereich 56944 – 56959, betreiben. Das heißt: 5 Steckplätze weiter nach rechts gehen = 5x16 = 80. Aus dem obigen Befehl A = Peek (56864) wird jetzt 56864 + 80 A = Peek (56944).
Beim Verschieben der Karte vom äußeren rechten nach dem äußeren linken Steckplatz (BU 2) wird 7x16 = 112 abgezogen. Aus dem angeführten Beispiel resultiert folgende Faustregel:
So oft man mit den Steckplätzen nach rechts geht – so oft muß man auch die Adresse um 16 erhöhen.

Ebenso geschieht dies umgekehrt; so oft man mit den Steckplätzen nach links geht, (BU 2) Enable 0, so oft muß man auch die Adressen um 16 verringern.

Wichtig: Alle übrigen Adressen, die in das Betriebssystem zeigen (Abfrage der Tastatur usw.), dürfen nicht verändert werden, nur die 16 Adressen des Steckplatzes sind veränderbar.

Wegen Platzmangel auf der Netzteilkarte befinden sich die zwei Elcos C1 + C2, die die Betriebsspannung von + 30 Volt glätten, auf der Busplatine. Die Widerstände R1 – R14 befinden sich ebenfalls auf dem Bus (Termenierung), aber davon später.

Es müssen 7 Drahtbrücken auf dem Bus verlegt werden, und zwar die Punkte A – G zwischen den beiden Steckplätzen BU 1und BU 2 mit den einzelnen Punkten A – G für jeden Steckplatz BU 3 – BU 9, BU 2 ist schon auf dem Bus verbunden. Dies sind die ENABLE-Leitungen für die einzelnen Steckplätze.

Die Widerstände R 15, R 16 und die Drahtbrücke zwischen BU 9 und BU 10 müssen nicht unbedingt montiert werden. Man kann 2x1kΩ einlöten. Damit werden RESET und INTERRUPT (IRQ) auf +5Volt gehoben. Dies dient zur Störunterdrückung.

Bei der Bestückung ist keine besondere Reihenfolge nötig, außer der Drahtbrücke A. Sie sollte vor der Montage des Kondensators C2 verlegt werden.

Die Buskarte sollte am besten in einem Einschubgehäuse montiert werden. Alle verwendeten Steckkarten sind in Europaformat 10x16 cm gehalten. Bei dem Steckplatz BU 1 ist zur linken Seitenwand ein Abstand von 30 mm und bei dem rechten Steckplatz BU 10 zur rechten Seitenwand ein Abstand größer als 30 mm zu lassen. Von Buchsenleiste zu Buchsenleiste besteht ein Rastermaß von 30 mm.

Achtung: Bei der Herstellung der Platine ist darauf zu achten, daß die Folie beim Belichten seitenverkehrt (spiegelbildlich) auf dem Fotolack aufliegt (Beschriftung darf nicht lesbar sein). In der nächsten CPU-Ausgabe erscheint als Bauanleitung die C-64-Buskarte sowie die Bescheibung der VC-20/C-64-Platinenstecker.

Tabelle 1

Buchsenleisten belegung Bus Bu 1 Linker Steckplatz Bus Karte

1	0V
2	A0
3	A1
4	A2
5	A3
6	D0
7	D1
8	D2
9	D3
10	D4
11	D5
12	D6
13	D7
14	ENABLE $\overline{0}$
15	S02
16	CRW
17	$\overline{IRQ}$
18	$\overline{RES}$
19	Frei
20	+5V
21	ENABLE $\overline{0}$
22	" 1
23	" 2
24	" 3
25	" 4
26	" 5
27	" 6
28	" 7
29	Frei
30	"
31	0V

Buchsenleisten belegung Bus Bu 2 – Bu 9 8 mittlere Steckplätze

1	0V
2	A0
3	A1
4	A2
5	A3
6	D0
7	D1
8	D2
9	D3
10	D4
11	D5
12	D6
13	D7
14	ENABL
15	S02
16	CRW
17	$\overline{IRQ}$
18	$\overline{RES}$
19	Termenierung
20	+5V
21	−5V
22	+15V
23	−15V
24	0V
25	+30V
26	+30V
27	~24V } a
28	~24V } a
29	~24V } b
30	~24V } b
31	0V

Buchsenleisten belegung Bus Bu 10 Rechter Steckplatz Netzteil

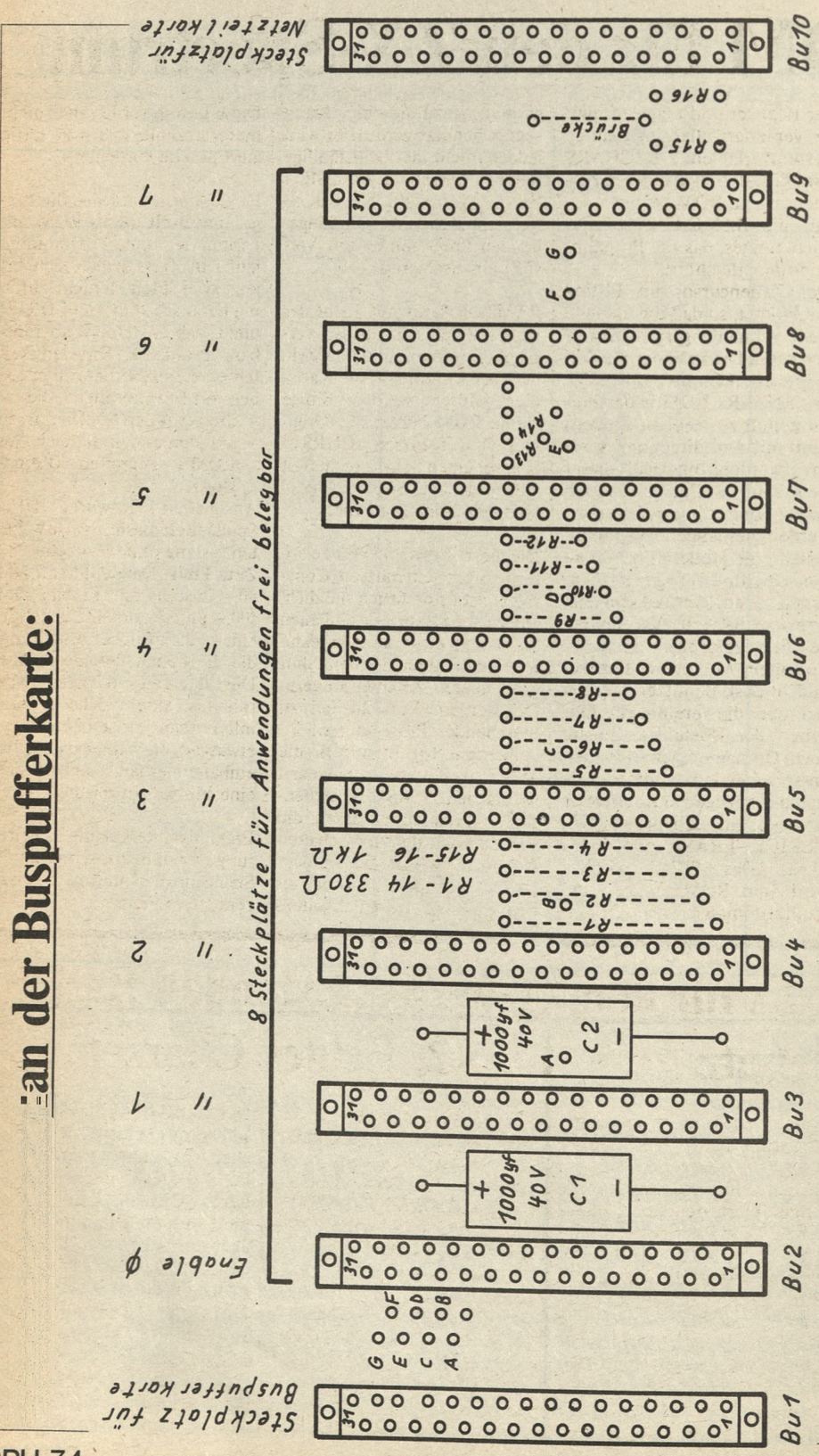
1	0V
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	Termenierung
14	
15	
16	+5V
17	+5V
18	−5V
19	+15V
20	−15V
21	0V
22	} +30V
23	} +30V
24	
25	} ~24a
26	} ~24a
27	
28	} ~24b
29	} ~24b
30	}
31	0V

Tabelle 2

Adressenbelegung der 8 Steckplätze

		DEZIMAL	HEXADEZIMAL
VC64	BU2	56832-56847	DE00-DE0F
	BU3	56848-56863	DE10-DE1F
	BU4	56864-56879	DE20-DE2F
	BU5	56880-56895	DE30-DE3F
	BU6	56896-56911	DE40-DE4F
	BU7	56912-56927	DE50-DE5F
	BU8	56928-56943	DE60-DE6F
	BU9	56944-56959	DE70-DE7F

		DEZIMAL	HEXADEZIMAL
VC20	BU2	38912-38927	9800-980F
	BU3	38928-38943	9810-981F
	BU4	38944-38959	9820-982F
	BU5	38960-38975	9830-983F
	BU6	38976-38991	9840-984F
	BU7	38992-39007	9850-985F
	BU8	39008-39023	9860-986F
	BU9	39024-39039	9870-987F



An alle Interessierten!

Der hier vorgestellte Bestückungsplan ist aus technischen Gründen nur im verkleinerten Format abgedruckt. Sowohl Bestückungsplan als auch der zum Selbstbau notwendige Platinenplan können beim TRONIC-Verlag gegen einen Unkostenbeitrag bestellt werden.
Falls Sie zu unserer neuen Hardware-Serie spezielle Fragen haben, wenden Sie sich bitte direkt an den Autor H. Thöne, dieser gibt Ihnen gern Auskunft.
(Di. u. Do. 18-19 Uhr / Tel. 05676-1376).

ISO-ROM für den ZX-Spectrum

Was der „Kleine“ alles in seinem ROM hat, überzeugte schon viele. Er kann einiges, wovon andere Computer nur träumen. Trotzdem: es war noch Platz, ca. 1 KByte haben die Konstrukteure freigelassen, vielleicht für die Microdrive-Steuerung (die dann schließlich doch nicht paßte)?
Nun, egal, warum noch Platz war, man kann ihn nutzen. Natürlich kann man nicht einfach etwas in das ROM schreiben, sondern muß den Inhalt des ROM's und die zusätzlichen Informationen in ein EPROM schreiben, das dann statt dessen eingebaut wird.
Herr Volker Marohn hat etwas derartiges getan. Er bietet ein neues ROM für den Spectrum an, das über interessante Fähigkeiten verfügt.
Die wohl für viele wichtigste Einrichtung ist ein kleines Knöpfchen, das einen Warmstart ermöglicht. Damit können in Endlosschleifen steckende Computer erlöst werden, ohne daß das Programm verloren geht.
Für die MC-Programmierer ist ein zweites Knöpfchen interessant: damit wird ein HEX-Monitor aufgerufen, der so ganz nebenbei auch die Inhalte einiger Register und Flags anzeigt. Er verändert allerdings einige Systemvariablen, z. B. CHARS wird grundsätzlich auf den Zeichensatz im ROM zeigen.
Für die BASIC-Fans gibt es auch einiges, das die Programmierung erleichtert.
Der Zeilencursor im Listing blinkt jetzt, so daß die aktuelle Zeile schneller gefunden werden kann. Die zu ändernde Zeile braucht nicht mehr mit LIST nr<ENTER> EDIT in den Editor geholt zu werden, sondern kann mit # nr direkt angesprochen werden. Innerhalb dieser Zeile kann der Cursor nun auch hinauf und hinunter geführt werden, bei langen Programmzeilen eine große Hilfe! Zu allem Überfluß springt der Cursor bei einem Syntax-Error jetzt direkt an die Stelle des blinkenden Fragezeichens.
Vor allem Microdrive-Besitzer und ß-Disk-Benutzer werden sich über die vereinfachte Eingabe der Befehle freuen. !LOAD “name“ generiert den entsprechenden Interface-1-Befehl, bezogen auf Drive 1. Analoges gilt auch für MERGE, VERIFY, ERASE und FORMAT. Das ßjDisk-Interface wird vom ROM erkannt und dann automatisch das DOS aufgerufen, wenn die obige Kurzsyntax benutzt wird. Hier wird jedoch nicht der vollständige Befehl (RANDOMIZE USR...: REM ...) generiert, so daß diese Vereinfachung in weiterzugebenden Programmen mit Vorsicht einzusetzen ist.

RAMTOP kann mit Hilfe des neuen Befehls $ adresse verschoben werden, ohne daß wie bei CLEAR adresse die Variablen gelöscht werden. Einige kleine ROM-Fehler des Originals sind behoben: CHR$ 9 (Cursor einen Schritt rechts) ist jetzt druckbar, SCREEN$(x,y) uneingeschränkt nutzbar.

Da diese EPROM's Einzelanfertigungen sein müssen (Copyright), ist es natürlich möglich, auch eigene Routinen einbauen zu lassen; diese können dann mit einem über Symbol Shift erreichbaren Zeichen aufgerufen werden. Vorsicht bei zu tauschenden Programmen!
Außerdem hat man z. B. die Möglichkeit, sich einen eigenen (deutschen??) ASCII-Zeichensatz einbrennen zu lassen, die Farben nach dem Einschalten (schwarze Schrift auf weißem Grund) zu ändern und noch einige andere Möglichkeiten. Diese Extras werden auch nicht teurer bezahlt, sofern sie fertig auf Cassette vorliegen.

Es stellt sich natürlich die Frage, inwieweit dieses ROM mit Peripherie und Programmen läuft. In Tests haben sich bis jetzt zwei Spiele als nicht lauffähig herausgestellt, mit FORTH und C gab es kleinere, aber lösbare Probleme. Schwierigkeiten ernsterer Natur wird es geben mit Erweiterungen, die
- ihrerseits den NMI benutzen
- auf den freien Bereich des ROM's zugreifen (Kempston)

Ansonsten: Tasword, DFü, Spiele, Beta Basic, usw. usw. liefen bislang ohne Probleme.
Zum Preis: Das ROM soll DM 80,- kosten, mit Einbau DM 100,- plus Porto. Dazu gehört eine ausführliche Anleitung, die die neuen Befehle erklärt. Die Lieferzeit soll lt. Herrn Marohn ca. eine Woche dauern, mit Einbau je nach Nachfrage etwas länger. Will man es selbst einbauen, erhält man natürlich eine Einbau-Anweisung.

Ergebnis: interessante Erweiterung der ohnehin schon guten Spectrum-Leistungen zu akzeptablem Preis.

Tips und Tricks für den TI-99/4 A Extended Basic

1. Listschutz

```
100 ON BREAK NEXT
110 CALL CLEAR ::
    INPUT CODE$
120 IF CODE$="TEST"
    THEN 1000
130 ON ERROR 150
140 RUN ""
150 RUN
1000 REM Hier beginnt das
     eigentl. Programm
```

Das Programm wird mit SAVE "CS1", PROTECTED abgespeichert. Es kann nun nicht mehr gelistet werden. Durch den Befehl in Zeile 100 kann das Programm auch nicht mehr gestoppt werden.

In Zeile 110 wird der Benutzer aufgefordert, den Code einzugeben. Wird das richtige Codewort eingegeben (hier "TEST"), dann verzweigt das Programm zum eigentlichen Programmbeginn in Zeile 1000.
Wird aber das falsche Codewort eingegeben, so ist das Programm verloren: Die Fehlermeldung "NO PROGRAMM PRESENT“ wird ausgegeben.

2. Echtes Charset

```
1000 SUB NEUDEF
1010 FOR Z=65 TO :: CALL
     CHARPAT(Z,A$)
1020 CHAR$="0000"&SEG$
     (A$,1,4)&SEG$(A$,7,4)
     &SEG$(A$,13,4)
1030 CALL CHAR(Z+32,
     CHAR$) :: NEXT Z
1040 CALL CHAR(99,"000000
     3C4040403C",106,"000000
     0808084830000000024283
     02824")
1050 SUBEND
```

Der Befehl "CALL CHARSET" soll beim TI eigentlich alle Zeichen in Ihre ursprüngliche Form zurückdefinieren, falls sie durch "CALL CHAR" umdefiniert wurden. Dies funktioniert aber nur bei den Großbuchstaben. Die Kleinbuchstaben kann man nun durch "CALL NEUDEF" ebenfalls in ihrer ursprünglichen Form definieren.

Tips und Tricks für den SVI-328

1. Bekanntlich läßt sich mit dem WIDTH-Befehl die Spaltenzahl des Bildschirms einstellen. Leider sind jedoch nur die Werte 39, 40 und, mit 80-Zeichen-Karte, 80 erlaubt, also zum Beispiel WIDTH 40 für die Darstellung von 40 Zeichen pro Zeile.

Will man aber nur 30 Zeichen darstellen, so ist dies über einen kleinen Trick möglich:
In der Speicherzelle &H F543 steht der Wert der Spaltenzahl. Da diese Adresse im RAM-Bereich liegt, läßt sie sich natürlich auch beschreiben.
Für die gewünschte Spaltenzahl 30 muß man also nur POKE -2749,30 eingeben, oder POKE -2749,20 wenn nur 20 Zeichen je Zeile gewünscht werden.

Im Gegensatz zum WIDTH-Befehl wird hier der Bildschirm nicht gelöscht. Dies kann von Vorteil sein, will man z. B. am rechten Rand des Bildschirms eine dauerhafte Anzeige haben, die vom normalen Bildschirm-Scrolling nicht betroffen wird. Zu beachten ist allerdings, daß CLS den gesamten Bildschirm löscht, also auch diejenigen Bildschirmpositionen, die mit PRINT nicht mehr ansprechbar sind.

Mit VPOKE sind nach wie vor alle Bildschirmpositionen ansprechbar.

2. Automatisches CAPSLOCK
Bei Eingaben ist es oftmals von Vorteil, wenn diese ausschließlich in Großbuchstaben erfolgen. Zwar kann man vor der Eingabe einen entsprechenden Hinweis ausdrucken oder umständlich alle Kleinbuchstaben wieder in Großbuchstaben umwandeln.
Eleganter ist jedoch, mit POKE -456,1 auf Großschrift umzuschalten. Mit POKE -456,0 schaltet man wieder auf Klein-/Großschrift um.
Die Adresse -456 = &H FE38 ist ein Flag, das Auskunft darüber erteilt, ob die CAPSLOCK-Taste gedrückt ist (Inhalt dann 1) oder nicht (Inhalt 0).

3. WELCHE TASTE IST GEDRÜCKT?
Im Bereich &H FD75 ... &H FD8A = -651 ... -630 ist die gedrückte Taste (bzw. Tasten) codiert. Dabei kommen jeder Adresse bestimmte Tasten zu:

FD75:	1!	20	3#	4$	5%	ô	7&	0)
FD76:	8*	9(	=+	""	:;	,<	.>	/?
FD77:	—	E	A	D	F	G	B	C
FD78:	I	O	L	K	J	H	N	M
FD79:	P	U	T	R	W	Q	S	V
FD7A:	↑	←	{[	}]	~\	X	Y	Z

FD7B: STOP ENTER CTRL SHIFT LEFTGRPH RIGHTGRPH ESC ←
FD7C: F1/F6 F2/F7 F/3F8 F4/F9 F5/F10 CLS INS
FD7D: DEL SELECT PRINT SPACE → ⇨
FD7E: 1 2 3 4 5 6 7 0 (Im Zahlenfeld)
FD7F: / * + - . , 8 9
FD80: wie FD75
FD81: wie FD76
FD82: wie FD77
FD83: wie FD78
FD84: wie FD79
FD85: wie FD7A
FD86: wie FD7B
FD87: wie FD7C
FD88: wie FD7D
FD89: wie FD7E
FD8A: wie FD7F

Beispiele:
Abfragen der SHIFT-Taste; Ausführung des Programms nur, wenn diese gedrückt ist:

```
.
.
100 if peek(&hfd7b)<>254 then 100
.
.
.
```

Auch alle Kombinationen mit den gedrückten Tasten sind möglich:
Tasten I O L M sind gedrückt ⇨ PEEK(&H FD78)=77.
Ist keine Taste gedrückt, dann ist der Inhalt der entsprechenden Speicherzelle = 255.
Es ist nun also möglich, auch den Tasten SELECT und PRINT, die ja keinerlei Bedeutung haben, eine Aufgabe zukommen zu lassen.
4. Es gibt eine einfache Möglichkeit zu überprüfen, ob eine Melodie gespielt wird oder nicht. Die Abfrage geschieht mit IF PLAY (...), wobei in der Klammer der Tonkanal (1, 2 oder 3) steht. Mit IF PLAY(0) werden alle Tonkanäle abgefragt.
Beispiel:

```
.
.
.
100play a$,b$,c$
.
.
.
```

Erst wenn die Melodie zu Ende ist, wird in der Programmausführung fortgefahren.
5. VPOKE ...,191
191 ist der Code für den Cursor.
VPOKE 0,191 bewirkt zum Beispiel, daß am linken oberen Rand des Bildschirms stets das Zeichen angezeigt wird, das sich an der Stelle des Cursors befindet, gleichgültig, wo der Cursor positioniert ist.

Wohin mit Maschinenprogrammen auf dem CPC 464?

Die Zusammenarbeit von Maschinensprache und Basic ist auf vielen Computern eine problematische Sache. So auch beim Schneider CPC 464: Hier muß man sich überlegen, wohin die Maschinenprogramme im Speicher gelegt werden sollen.
Vom Hersteller Amstrad wurde vorgesehen, die Maschinenprogramme direkt unter dem HIMEM abzulegen und dann mit MEMORY diese Speichergrenze nach unten zu verschieben. Diese Methode verursacht aber eine Fehlermeldung beim nächsten SYMBOL AFTER, wenn man also eigene Grafikzeichen verwenden will. Der Grund ist recht einfach: Die Zeichen liegen zusammen mit dem Maschinencode am oberen Speicherende. Um zu verhindern, daß sich beide gegenseitig beeinflussen, quittiert der Basic-Interpreter die Verwendung von SYMBOL AFTER mit einer Fehlermeldung, wenn vorher mit MEMORY etwas am Speicherende verändert wurde.
Im folgenden werden einige Möglichkeiten beschrieben, Maschinenprogramme an verschiedenen Stellen im Speicher unterzubringen:

1. Direkt unter HIMEM
Wie beschrieben, gibt es Probleme, wenn Zeichendefinitionen benutzt werden. Abhilfe: Vor dem Laden des Maschinenprogramms muß SYMBOL AFTER O eingegeben werden. Dann können trotz Maschinensprache alle Zeichen definiert werden. Sollen allerdings SYMBOL und MEMORY in einem Programm untergebracht werden, und das Programm soll mehrmals laufen, z. B. bei der Programmentwicklung, so muß man sich nach anderen Speicherbereichen umsehen.

2. In Stringvariablen
Kurze Maschinencode-Routinen bis maximal 255 Bytes können in Stringvariablen abgespeichert werden. Dazu werden diesen Variablen die Maschinenbefehle als ASCII-Codes zugewiesen, z. B.
a$=CHR$(&CD)+CHR$(&6C)+CHR$(&BB)CHR$(&C9)
Dies ist eine Maschinenroutine, die den Bildschirm löscht. Aufgerufen wird das Programm mit dem Variablenzeiger @, dem Klammeraffen, entsprechend obigem Beispiel also:

CALL@a$

Hier gibt es aber eine Einschränkung: Da der Basic-Interpreter bei der Garbage Collection die Strings beliebig im Speicher verschieben kann, dürfen innerhalb des Maschinenprogramms keine absoluten Sprünge wie JP oder CALL auftauchen.

3. In der Funktionszeichentabelle
Auch diese Methode eignet sich nur für kurze Routinen, denn dieser Speicherbereich ist nur rund 100 Zeichen groß. Dafür verschiebt sich der Speicher nicht, es können also beliebige Maschinenbefehle benutzt werden. Die Adresse des Buffers erhält man mit:

PRINT PEEK (&B4E1)+PEEK(&B4E2)*256
Das liefert normalerweise den Wert 46150. Ab dieser Adresse können Maschinenprogramme eingePOKEd werden.

```
10 KEY DEF 15,0,48,48,48
20 KEY DEF 13,0,49,49,49
30 KEY DEF 14,0,50,50,50
40 KEY DEF 5,0,51,51,51
50 KEY DEF 20,0,52,52,52
60 KEY DEF 12,0,53,53,53
70 KEY DEF 4,0,54,54,54
80 KEY DEF 10,0,55,55,55
90 KEY DEF 11,0,56,56,56
100 KEY DEF 3,0,57,57,57
110 KEY DEF 7,0,46,46,46
120 KEY DEF 6,0,13,13,13
```

Damit wird der ganze Zehnerblock mit Zahlenwerten belegt und benötigt keinen Speicher mehr für die Funktionstasten.
Es ist dabei zu beachten, daß keine Funktionstasten mehr gedrückt werden, da das unabsehbare Folgen haben kann. Am besten ist es, die Funktionstasten als Zehnerblock zu benutzen und damit auf die Stringbelegung mit KEY zu verzichten. Mit den folgenden Befehlen wird das sichergestellt:
CALL &BBOO (Initialisiert die Tastatur und sperrt BREAK, sollte deswegen nur im Direktmodus verwendet werden)

4. Unterhalb des Basic-Speichers
Dies ist die wohl universellste Methode, da alle bisherigen Einschränkungen vermieden werden können: SYMBOL AFTER kann ohne Probleme benutzt werden, der Maschinencode muß nicht relokatibel (verschiebbar) sein, und es gibt keine Beschränkung des Speicherplatzes. Es kann der gesamte für Basic nutzbare Speicher dem Maschinenprogramm zur Verfügung gestellt werden.
Um diese Methode zu verstehen, muß man einiges über die interne Speicherung von Basic-Programmen wissen: Der Interpreter speichert sie ab Adresse 370 (&0172) ab; die Adressen &AE81 und &AE82 enthalten Zeiger auf diesen Basic-Beginn. Verändert man diese Zeiger, so kann Speicherplatz reserviert werden. Es ist wichtig, diese Zeiger im Direktmodus umzuPOKEn, da sich ein Programm durch die POKEs selbst löschen würde, bevor es damit fertig ist. So wird der Speicherplatz vor Basic geschützt:
Laden Sie die Variable LOW mit der letzten zu schützenden Adresse und geben sich folgendes ein:

```
LOW$=HEX$(LOW,4):POKE &AE82,VAL
('&'+LEFT$(LOW$,2)):
POKE &AE81,VAL('&'+RIGHT$(LOW$,2)):NEW
```

Der NEW-Befehl ist notwendig, um sicherzugehen, daß auch die Variablenzeiger und andere Pointer richtiggestellt werden. Nun können Maschinenprogramme, die im Binär-Format vorliegen, ganz einfach mit LOAD ab Adresse 370 geladen werden. Natürlich kann das Programm auch mit DATA-Zeilen eingePOKEd werden.
So, jetzt dürften Sie keine Schwierigkeiten mit der Unterbringung Ihrer Maschinenprogramme mehr haben. Viel Spaß beim Programmieren!

Martin Kotulla

Allgemeine Hilfen zum VC-20-Computer

A. „Diese verflixten Steuerzeichen . . .!"
Ein Stück schwarzes Papier, aus dem man ein Quadrat in der Größe der abgedruckten Zeichen geschnitten hat, um so eine Art „Fenster" zu machen, ist hier sehr nützlich. Mit Hilfe dieses „Rahmens" lassen sich die invertierten Steuerzeichen sowie Grafikzeichen besser voneinander trennen und erkennen.

B. Die Programmlänge.
Um die Länge eines im Speicher befindlichen Programmes zu ermitteln, gibt man im Direktmodus folgende Zeile ein:

```
FOR L = (PEEK(43)+256*PEEK(44)) TO (PEEK (55)+
256*PEEK(56)):
IF PEEK(L) OR PEEK(L+1) THEN NEXT:
L=L-(PEEK(43)+256*PEEK(44)):PRINT L
```

L ist die Länge des Programmes.
Es empfiehlt sich, die Länge sowohl VOR als auch NACH einem Programmlauf zu ermitteln, insbesondere bei längeren Programmen, damit man einen Überblick über den gesamten Speicherplatz bekommt.

C. Speicherplatz sparen.
Folgende BASIC-Befehle benötigen die INT-Funktion NICHT, da INT im System ROM automatisch ausgeführt wird:

```
CHR$ LEFT$ MID$ RIGHT$ PEEK POKE GOTO
GOSUB SPC TAB DIM
```

Dasselbe gilt für indizierte Variablen.

D. Dokumentationslücke.
Vier Grafikzeichen werden beim VC 20 häufig übersehen, weil sie nicht auf den Tasten markiert sind. Es handelt sich um Zeichen, die nur im Textmodus (Zeichensatz 2 – durch Drücken von C= + SHIFT) erreichbar sind. Die Zeichen und deren numerische Werte erscheinen lediglich in der Bildschirm-POKE-Tabelle im VC-20-Handbuch.

TASTE	POKE	CHR$
SHIFT + Pi	94	126/222
SHIFT + £	105	169/233
SHIFT + @	122	186/250
C= + *	95	127/223

E. DATA-Fehler einkreisen.
Die Zeilennummer einer fehlerhaften DATA-Eingabe läßt sich mit folgender Eingabe ermitteln:

```
PRINT PEEK(63)+256*PEEK(64)
```

F. „Verdoppelte" Disketten: eine WARNUNG!
Die Umwandlung von einseitigen in beidseitige Disketten wird häufig als Sparmaßnahme angepriesen; ferner wird behauptet, daß eine formatierte Diskette dieser Art keine Probleme mit Daten verursachen würde. Dabei wird aber etwas Wesentliches immer verschwiegen: bei der Benutzung von – auch echten – beidseitigen Disketten in einem „nur einseitigen" Laufwerk, z. B. VC1540/1541, laufen die beiden Seiten in entgegengesetzter Drehrichtung! DER ABRIEB VON DER DISKETTE WIRD BEIM UMDREHEN DER DISKETTE MÖGLICHERWEISE DURCH DIE VERÄNDERTE DREHRICHTUNG WIEDER AUF DIE DISKETTE GETRAGEN – WAS DANN ZU DATENVERLÜSTEN FÜHRT.
Wer dieses Problem bereits erlebt hat, ohne eine Erklärung zu finden, erkennt jetzt, was er falsch gemacht hat.

G. Anmerkungen zu Zufallszahlen,
Bei Spielen werden sehr oft Zufallszahlen benutzt? aber sind sie immer „zufällig"? Probieren Sie folgendes:

```
10 FOR A=1 TO 6: B=INT(RND(1)*100):PRINT B:NEXT
```

Notieren Sie die Werte, schalten Sie den Computer aus und dann wieder ein – kein Reset. Jetzt wiederholen Sie die Zeile 10. Wenn Sie das richtig gemacht haben, erscheinen die gleichen Zahlen wieder. Die „Zufälligkeit" läßt sich erhöhen, wenn man den Wert (1) durch (0) ersetzt. Die Übereinstimmung der Zahlen kommt daher, daß beim Einschalten des Computers die Speicherzellen 139 bis 143 (dezimal) vom Betriebssystem „gesetzt" werden, und da es sich um eine ROM-Routine handelt, sind es immer dieselben Werte, die in diesen Zellen abgelegt werden.
Die in diesen Zellen „generierten" Zahlen sind auch nicht alle gleichwertig. Aus 139 kommen nur die Zahlen von 114 bis 128, wobei die nächst höhere Zahl etwa doppelt so oft vorkommt wie die vorangegangene. 140 liefert nur die Zahlen von 0 bis 127, nur die letzten drei Zellen liefern Werte von 0 bis 255.
Aus diesen Gründen erreicht der BASIC-Befehl RND erst nach einer großen Anzahl von Werten eine annähernd „zufällige" Verteilung. Eine andere Möglichkeit, „Zufallszahlen" zu erhalten, ist durch PEEKen der Jiffyzahl in der Systemuhr, die innerhalb von etwa vier Sekunden alle Zahlen von 0 bis 255 durchläuft. Dies geschieht mit: PEEK(162). Wenn man die obere Grenze der Zahlenreihe bestimmen will, geht das mit

```
PEEK(162) AND X
```

wobei X der höchste zulässige Wert ist. Wer nur gerade Zahlen haben will, erreicht das mit:

```
PEEK(162) AND 254
```

Folgendes kurzes Programm erzeugt B-Zahlen aus einer Anzahl von A-Elementen:

```
10 N = 1: DIM P(A)
15 X = INT(RDN(0)*A+1):IF N > 1 THEN 100
20 P(N) = X: PRINT P(N): N = N+1
25 IF N = B + 1 THEN END
100 FOR M = 1 TO N: IF X = P(M) THEN GOTO 15
110 NEXT M: GOTO 20
```

Vorankündigung: Hessische und Bayrische Computertage

Der im November 1984 stattgefundene 1. Bayrische Computer-Tag findet in diesem Jahr in Form des 2. Münchener Computer-Tages seine Fortsetzung. Der 2. Münchener Computer-Tag wird am 24. November im Löwenbräukeller durchgeführt.
Die 7. Hessischen Computer-Tage finden am 9. und 10. November in Neu-Isenburg bei Frankfurt, in der Hugenottenhalle, statt.

Setzen Sie zum Endspurt an . . . Software-Champion 1985!!!

Es warten tolle Preise auf Sie!

1. PREIS:

5000 DM in bar!

2. PREIS:

Ein Farbmonitor im Wert von 800,– DM!

3. PREIS:

Computer-Zubehör (wahlweise im Gesamtwert von 300,– DM)

Jedes eingesandte Programm nimmt an der Auswahl teil!!!
Letzter Annahmetermin ist der **30. September 1985.** Der Rechtsweg ist ausgeschlossen.
Nutzen Sie Ihre Chance, beteiligen Sie sich mit einem Software-Programm ab unserem Wettbewerb!

Zur Auswahl:
Wir stellen in jeder Ausgabe von COMPUTRONIC mindestens drei TOPPROGRAMME, in „HOMECOMPUTER" 1 Topprogramm vor. Hierfür trifft die Redaktion die Entscheidung (unabhängig vom Computertyp).
Wird ein Programm zur Darstellung im Heft herangezogen, zahlt der Verlag ein Honorar von DM 120,– pro abgedruckte volle Seite. Zum Jahresende stellt die Redaktion noch einmal alle als TOPPROGRAMME ausgezeichneten Programme vor. Unsere Leser und die Redaktion werden dann in der Ausgabe Nov./Dez. '85 das absolute TOPPROGRAMM des JAHRES wählen.
Der Autor des Programmes wird in unserer Ausgabe vorgestellt und mit DM 5000,– prämiiert werden.

Es winken irre Preise

Einzusenden sind:
++ Spielbeschreibung
++ Datenträger
++ Listing (nicht unbedingt erforderlich)

Einsenden an: TRONIC-VERLAG – Postfach – 3444 Wehretal 1
KENNWORT: SUPER-SOFTWARE-CHAMPION

Bitte frankieren Sie Ihren Freiumschlag mit 1,40 DM!

Wir wünschen allen Freizeitautoren viel Spaß beim Mitmachen.

CPU-Computer-Club-Vorstellung

Der 1. VZ.-/Laser Fan-Club

Im Februar 1984 wurde der 1. VZ 200 Fan-Club gegründet, der auch den Besitzern des Laser 110/210 offensteht.
Die Absicht des Clubs ist es, alle greifbaren Informationen zu sammeln und in Form einer 2-monatlich erscheinenden Club-Info allen Clubmitgliedern zu übersenden.
Die bisher erschienenen Club-Infos haben bei den Mitgliedern immer ein positives Echo gefunden, da sie viel Neues enthielten. Die Hauptaufgabe des Clubs soll es sein, die von den Mitgliedern erstellten Programme den anderen Mitgliedern zugänglich zu machen. Der Tausch läuft bisher bei der hiesigen Clubzentrale reibungslos ab.
Das Wichtigste für alle Mitglieder ist es, mitzuarbeiten.?!!
Mit Ihren Beiträgen über zb.
Tips und Tricks –
Computerrätsel –
Grafik –
Maschinensprache –
Programmierhilfen
und was es sonst noch so alles gibt, um mit dem VZ 200 besser zu arbeiten, sollen Sie helfen, die Club-Infos wirklich attraktiv zu gestalten.
Sie können auch kostenlos! Inserate in der Info einsetzen –
wie – Suche nicht computergestörte Weiblichkeit, biete im Tausch guten Kleinrechner
oder – Verkaufe Speichererweiterung für VZ 200 – 16 KB
oder auch konkrete Fragen stellen.
Hinsichtlich des Programmtausches wäre noch zu sagen, daß diese möglichst auf Kassette oder Diskette ggf. auch als Listing an Wolfgang Maschke übersandt werden müßten. Dort werden die Programme dann kopiert und bei Nachfrage an die Mitglieder versandt.
Zu den Programmen gehört natürlich auch eine Beschreibung und die Angabe der erforderlichen Optionen, damit eine Liste der zur Verfügung stehenden Programme erstellt werden kann.
Die Mitgliedschaft ist kostenlos, allerdings wird um einen Clubeinstand von 20,– DM gebeten, der bei Herausgabe der 2-monatlich erscheinenden Club-Info verrechnet wird.
Wenn Sie Mitglied des 1. VZ./Laser Fan Club werden wollen, dann übersenden Sie den Betrag von 20,– DM + Beitrittserklärung an:

Wolfgang Maschke
Waltersdorfer Chaussee 99
1000 Berlin
Postgirokonto Berlin-West
Nr. 36460-107

Sie erhalten dann umgehend die gewünschten Club-Infos.

Erweitertes Angebot für die Mitglieder des Sinclair Benutzer Clubs

- Monatlich erscheint unser Clubinfo mit ca. 16 Druckseiten (entspricht ca. 64 DIN-A4-Seiten Text), Listings, Tests, Hardware, kostenlosen Anzeigen usw. Die Anzeigen werden auch in unserer MB und in den Info's anderer Clubs gebracht.
- Die Clubmitglieder können mit dem Club oder auch untereinander zahlreiche, von den Autoren freigegebene Programme tauschen.
- Die Bildung von Ortsgruppen ist geplant, sofern genügend Interesse besteht.
- Es finden in unregelmäßigen Abständen Clubtreffen statt. Für die von Zeven weiter entfernt Wohnenden gibt es teilweise die Möglichkeit, bei Treffen befreundeter Clubs zu Gast zu sein.
- Der Club hat eine wachsende Leihbücherei. Diese Bücher können für maximal 4 Wochen ausgeliehen werden.
- Der Club hat vergünstigte Einkaufsmöglichkeiten; die Rabatte betragen bis zu 40% (keine Sammelbestellungen!). Derzeit sind es vier Händler: Müller hard & Soft, Flensburger Computerversand, Uwe Fischer und Bauz GmbH.
- **Druckservice für alle, die Ausdrucke benötigen. Derzeit stehen sechs verschiedene Drukker zur Auswahl, die verschiedene Möglichkeiten bieten.**
- Mitglieder haben diverse Erweiterungen für ZX 81 und Spectrum zum Verleih zur Verfügung gestellt, z. B. Digitaltracer, Lightpen, Microspeech, ZX-Printer und anderes.
- **Die Wicking-Box hat für den SBC eine geschlossene Usergruppe eingerichtet. Zudem haben wir natürlich sowieso Zutritt zum allgemeinen Teil, der sehr Sinclair-freundlich eingestellten Box.**
- Ein Austritt aus dem Club ist jederzeit möglich. Monatsbeiträge, die schon im voraus entrichtet wurden, werden dann erstattet. Weitere Ansprüche bestehen nicht. Die Clubleitung behält sich vor, ein Mitglied auszuschließen, das grob gegen die Satzung des SBC verstößt.
- Die Aufnahmegebühr beträgt DM 10,–, der Monatsbeitrag DM 5,–.
- Interessenten wenden sich an: **Erika Hölscher, Ernst-August-Str. 5, 2730 Zeven Tel. 04281/6442**

Computer Club „List Soft“

LIST SOFT IST EIN CLUB DER ALLE JUGENDLICHEN BENUTZER DER COMPUTER TI99/4a UND VC-20 ANSPRECHEN SOLL.
WIR WOLLEN VERSUCHEN CA. JEDEN MONAT EINE ZEITSCHRIFT ZU ENTWICKELN AN DER SICH ALLE MITGLIEDER BETEILIGEN SOLLEN.
UNSERE CLUBZENTRALE IST IN HOF, ABER ES SOLLEN MITGLIEDER AUS GANZ DEUTSCHLAND IN DEM CLUB VERTRETEN SEIN.
DER CLUB BESTEHT ZUR ZEIT AUS 10 MITGLIEDERN.
DER CLUBBEITRAG BETRÄGT MONATLICH 3 DM.
VON DIESEM GELD WIRD DIE ZEITSCHRIFT FINANZIERT UND JEDEM MITGLIED GRATIS ÜBERSANDT.
WIR WOLLEN VERSUCHEN PROGRAMME, TIPS, TRICKS UND PROGRAMMIERTECHNIKEN AUSZUTAUSCHEN.

KONTAKTADRESSE:
MICHAEL ECKERT
ZOBELSREUTHERSTR. 30
8670 HOF (SAALE)
ODER
RALF HOPPERDIETZEL
TH.-STORM-STR. 22
8670 HOF (SAALE)

CPU-Podium ***

Benchmark-Test

Die Redaktion:

Wir möchten uns an dieser Stelle bei unseren Lesern – Peter Heupel aus Dietzhölztal und Thomas Mäurer aus Oberhausen – für die Ergänzung des Benchmark-Tests aus der CPU Nr. 7 bedanken.
Peter Heupel besitzt ein Sinclair QL-Computer und ermittelte in drei Versuchen folgende Zeiten:
Grundausführung 128 K:
Benchmark 1: 1,81 Sek.
Benchmark 2: 5,34 Sek.
Mit Speichererweiterung auf 38 K:
Benchmark 1: 1,44 Sek.
Benchmark 2: 4,70 Sek.
Aus diesem Ergebnis ergibt sich ein klarer Spitzenplatz im Benchmark-Test 1 und ein vierter bzw. zweiter Platz im Benchmark-Test 2.
Thomas Mäurer ist im Besitz eines MTX 500-Computers und ermittelte folgende Zeiten:
Benchmark 1: 5,9 Sek.
Benchmark 2: 8,0 Sek.

Computersysteme

Sehr geehrte Herren der CPU-Redaktion!

. . . Im weiteren muß ich Ihnen ein großes Lob aussprechen. Endlich mal eine Zeitschrift, die nicht immer nur „Spielchen“ bringt, sondern auch vernünftige Sachen. Erwähnenswert ist hier auch, daß endlich einmal Maschinenprogramme für den TI-99 abgedruckt werden, die dann auch das Prädikat „Besonders gut“ verdienen.

Antwort der Redaktion:

Das CPU-Team freut sich natürlich, daß vielen unserer Leser (dieser Brief wurde stellvertretend für zahlreiche Einsendungen abgedruckt) diese Anwenderprogramme gefallen. Zudem werden wir uns auch in Zukunft weiterhin darum bemühen, Programme von etwas „älteren“ Computersystemen wie beispielsweise TI-99, Apple 2 oder VC-20 zu veröffentlichen.

Die Gewinner von CPU-Happy aus Heft Nr. 7

Aus den zahlreichen Einsendungen konnten dieses Mal 19 Gewinner ermittelt werden.

1. Preis
10 Kassetten aus unserem Software-Service. Hartmut Weber, Heidenheim

2. Preis:
3 Kassetten aus unserem Software-Service. Michael Blümlein, Kaufbeuren

3. Preis:
2 Kassetten aus unserem Software-Service. Dietmar Schmid, Rielasingen 1

4.-19. Preis:
1 Kassette aus unserem Software-Service:
Günter Heitel, Bietigheim
Lothar Maluche, Lindenberg/Allgäu
Dirk Kolbe, Hilgert
Alois Leimüller, Köstendorf (Österreich)
Michael Beil, Backnang
Gregor Gellissen, Münster
Berthold Prause, Wendeburg
Wilfried Gerndt, Wenden 1-Altennof
Erika Hölscher, Zeven
Bernhard Kappler, Nufringen
Thomas Möller, Herne
Christian Liechti, Wädenswil (Schweiz)
Frank Perau, Schiffdorf-Spaden
W. Ganschinietz, Thierhaupten
Alfred Rehberger, Feldheim
Peter Poellesch, Landshut

Neuer Autoren-Wettbewerb in der CPU!

Das CPU-Team startet gleich zu Beginn der „erweiterten Ausgabe“ einen tollen Wettbewerb!

Tolle Gewinne warten auf Ihre Gewinner: Insgesamt **600 Deutsche Mark** *werden jeden Doppelmonat für die besten Autoren-Beiträge „an den Mann gebracht“. Die Wettbewerbs-Kriterien: Der Wettbewerb gilt für die besten Einsendungen in den Rubriken*

1. User (maximal drei Seiten sollten bei uns „ankommen“)
2. Utility (hier sollten es max. zwei Seiten sein)
3. Tips & Tricks (hier maximal eine Seite)

Das Team der CPU wird die besten Einsendungen und Tips wie folgt honorieren:

für den besten „User-Text“ = 300 Mark,

für die beste „Utility“ = 200 Mark und

für die besten Tips & Tricks = 100 Mark

Sie sehen also, Mitmachen lohnt sich! Nur, wer mitwirkt, kann gewinnen. Und: Mehr noch – der „Spitzen-Autor“ wird in der nächstfolgenden Ausgabe der CPU vorgestellt.
Wenn Sie sich an diesem Wettbewerb beteiligen wollen, übersenden Sie uns bitte folgende Unterlagen:

1. **Ausführliche Beschreibung des eingesandten Beitrages**
2. **Datenträger (Kassette oder Diskette)**
3. **Komplettes Listing (wenn zur Beschreibung erforderlich)**
4. **Für die Autorenvorstellung: Infos zur eigenen Person (kurzer Lebenslauf, Hobbies etc.) + Foto**
5. **Frankierten Freiumschlag (mind. 1,40 DM)**

All dies senden Sie bitte an den TRONIC-Verlag, Postfach, 3444 Wehretal 1.

Also, mitmachen, mitwirken und mitgewinnen! CPU – das Magazin mit Pfiff!

Unterbrechungsfreie Stromversorgung

So schnell und genau Computer auch arbeiten mögen, so abhängig sind sie doch von einem wesentlichen Faktor: Von einer unterbrechungsfreien Stromversorgung. Doch so selbstverständlich ist es nicht, daß eine EDV-Anlage ständig mit der nötigen Energie versorgt wird. Jedes Einschalten eines anderen Verbrauchers kann einen kurzzeitigen Einbruch des Stromflusses auslösen – und damit große Schäden in einem Computerprogramm heraufbeschwören.

Die Oldenburger Firma **MICRO CONSULT** hat jetzt ein Gerät auf den Markt gebracht, das das scheinbar grundlose Abstürzen eines Computers verhindert: Die OUSV 180 R hält, zwischen Steckdose und EDV plaziert, die Spannung am Computer für mindestens zehn Minuten bei 180 VA Last aufrecht, nachdem ein Stromausfall eingetreten ist. Der USV 180 R eigene Rechteckausgang (220 Volt) hat gegenüber dem gebräuchlichen Sinusausgang die Vorteile eines höheren Wirkungsgrades, einer extrem niedrigen Abwärme und einer einfachen Regelung der Ausgangsleistung in weiten Grenzen. Zudem hat die USV 180 R die Vorteile, eine Umschaltzeit zu benötigen (im Gegensatz zu herkömmlichen umschaltenden Geräten), den Wechselrichter dauernd in Betrieb zu halten, die Ausgangsspannung immer netzsynchron zu halten und keine Phasensprünge zuzulassen. Die Unterschiede zu bisher bekannten Schutzgeräten – auch zu den vom Werk in die Computer eingebauten – sind also offensichtlich.

Zu beziehen bei:
Eckhard Koch
Carl-Eggers-Weg 7
2900 Oldenburg

DUO-PER/U-Universalumschalter

Mit dem Schnittstellenumschalter DUO/PER/U, betritt der Schnittstellenspezialist BAUZ aus Zeven Neuland. Der DUO/PER/U ist ein Schnittstellenumschalter auf Europakarten. Für jede anzusteuernde Schnittstelle wird eine Karte eingesetzt. Damit ist es möglich, den Umschalter und seinen Aufbau kundenspezifisch zu gestalten. Es kann damit von beliebig vielen auf beliebig viele Geräte umgeschaltet werden. Erwähnenswert ist noch die Möglichkeit, die Umschaltung am Gerät selbst durch Fernschaltung und per Softwaresteuerung vornehmen zu lassen.

Nähere Informationen bei:
BAUZ GmbH
Postfach 13 29
2730 Zeven
0 42 81 / 15 77

Floppyeinheiten zur PC-Erweiterung

Über den Distributionsvertrag mit der UPI Corp, einem multinationalen Peripheriegeräte-Lieferanten, kann die SCSE GmbH nun auch Floppyeinheiten zur PC-Erweiterung liefern. Das Subsystem, das über eine eigene Stromversorgung verfügt, ist mit einem oder zwei Floppy-Laufwerken lieferbar. Durch Selektion der Laufwerksadressen kann das Subsystem an IBM-PCs oder kompatiblen Computern mit einem oder auch zwei Floppy-Laufwerken benutzt werden, so daß maximal vier Floppy-Laufwerke zur Verfügung stehen.
Durch die Verwendung der Shugart-Standard-Schnittstelle SA 460 kann das Subsystem auch von OEMs eingesetzt werden.

Der Einzelstückpreis soll unter DM 2.000,– liegen.

Das Standard Basic-Buch zum Schneider CPC 464

Eine Edition aus dem Heim-Verlag, Darmstadt, ISBN 3-923250-12-6

Das Buch bietet methodisch strukturierte Grundlagen, die benötigt werden, um damit selbständig die ganze Palette des Programmierens anhand von ansprechenden Programmen zu erfahren. Ziel dieses Titels ist es, dem Leser einen Einstieg in die Arbeitsweise der Programmiersprache BASIC zu ermöglichen. Gezielte Problemstellungen werden in der für Computer verständlichen Sprache BASIC formuliert und in ein Programm umgesetzt.
Eine Programmiersprache zu beherrschen heißt, sie durch praktisches Anwenden am Computer zu erproben und begreifen zu lernen. Das sind die Ziele, die mit diesem Standard-BASIC-Buch angestrebt werden.
Die Programmbeispiele sind auf diese Zielrichtung abgestellt, so daß nicht allein fertige Programmlösungen geboten werden, sondern Programme, die die Kreativität fördern und die Platz für eigene Ideen lassen. Freiräume also für Erweiterungen jeglicher Art, weil nur dadurch die notwendigen Kenntnisse und Erkenntnisse möglich werden, die nicht allein durch „Abtippen“ oder „Laden“ irgendwelcher Übungsprogramme erlangt werden.

Trotzdem bietet das Buch Lösungen für bestimmte Problemstellungen an, wie z. B. eines der Projekt-Programme „Zeichendefinition“. Hier werden einerseits die Möglichkeiten des >>CPC 464<< aufgezeigt, andererseits dem Lernenden exemplarisch Programmierpraxis vermittelt, die er für sich selbst nutzen kann.
Einige Kapitel aus dem Inhalte:
- Die FOR-NEXT-Schleife
- Zeichenkettenverarbeitung in Tabellen
- Programmbeispiel zur Tabellenarbeit
- Sortieren von Zahlen und Zeichenketten
- Standardfunktionen
- Benutzerfunktionen
- Textverarbeitung mit dem CPC 464
- Das Arbeiten mit Dateien
- Fehlermeldungen und Fehlercodes
- Programmablaufpläne
- ASCCII-Werte Tabelle
- Die BASIC-Befehle
- Und einiges Mehr

Negativ bleibt bei diesem Werk festzuhalten, daß einige Listen und Texte aus dem Original-Schneider-Handbuch übernommen wurden. Positiv zu vermerken sind die recht zahlreichen kleineren und nützlichen Beispielprogramme.

Werden Sie Mitarbeiter der „CPU“

Die Redaktion sucht ständig neue Beiträge zur Veröffentlichung. Sollten Sie

- *Tips und Tricks für verschiedene Computersysteme*
- *kleine Maschinen-Routinen*
- *BASIC-Unterprogramme*
- *Hardware- und Software-Berichte*
- *sonstige interessante Beiträge rund um den Computer*

entwickelt haben, lassen Sie uns diese zukommen. Wir benötigen von Ihnen:

1. *Ausführliche Beschreibung des eingesandten Beitrages*
2. *Datenträger (Kassette oder Diskette)*
3. *Komplettes Listing (nicht unbedingt erforderlich)*
4. *ausreichend frankierter Rückumschlag (mind. 1,40 DM).*

Wird ein Beitrag zur Veröffentlichung herangezogen, zahlt Ihnen der Tronic-Verlag ein Autorenhonorar von 120,-- DM pro voll abgedruckte Seite.

WDR-Schulfernsehen bringt Sendebeiträge für Computerinteressierte

Bit und Byte: Wir bauen einen Computer

6 Folgen zu je 15 Minuten – 9./10. Schuljahr

Erstsendung	Mi.	17.45 Uhr
Wiederholungen	Do.	10.40 Uhr
	Fr.	8.10 Uhr
	Mo.	9.05 Uhr

	Erstsendung	
1. Jede Menge Chips	Mi.	28. 8. 1985
2. Aller Anfang ist leicht: Die Anzeige	Mi.	4. 9. 1985
3. Vergißmeinnicht: Der Speicher	Mi.	11. 9. 1985
4. Einer für alle: Der Prozessor	Mi.	18. 9. 1985
5. 001, das erste Programm	Mi.	25. 9. 1985
6. Befehle nach draußen: Die Peripherie	Mi.	2. 10. 1985

Ziel dieser Reihe ist es, die Funktionsprinzipien eines Computers und dessen Zentraleinheit, einen Mikroprozessor, an einem Beispiel zu erklären. Didaktisches Prinzip ist es, dies handelnd „begreiflich“ zu machen, indem ein Computer gebaut und programmiert wird.

Der Bau des Computers erfordert weder besondere Fähigkeiten noch spezielles Werkzeug: Lötkolben, Bohrer und Zange. Programme können beispielsweise zur Schaltalgebra geschrieben und eingegeben werden. Weiter können Geräte und Anlagen gesteuert werden, z. B. Fahrstuhl, Ampelanlage, Roboter etc.

Lehrerinformationen und kopierfähige Schülerarbeitsblätter zu dieser Reihe erscheinen in der Zeitschrift „Praxis Schulfernsehen“, Heft 109/110 und 111.

Dies ist was für scharfe Rechner!

CPU Bestellkarte-Softwareservice

Alle im Heft abgedruckten Programme können als zusätzlicher Service über den Verlag bezogen werden.
(Ausland nur gegen Vorkasse)

Die Zustellung erfolgt: gegen **Vorkasse** ☐ oder Inland per **Nachnahme** ☐ + Versandkosten

innerhalb von 1 Woche

Entnehmen Sie bitte aus unseren Preislisten die notwendigen Angaben für Ihre Bestellung:

Bitte liefern Sie mir

☐ Cassette für _____ System ☐ Anzahl

☐ Diskette für _____ System ☐ Anzahl

zum Preis von gesamt _____ DM

Name/Vorname: _____ Straße, Nr.: _____

PLZ/Ort: _____ Datum, Unterschrift

Abonnements:

wer jetzt in's Computerleben einsteigen will, sollte nicht überlegen: ein Abonnement ist immer eine lohnende Sache!

Ausschneiden und einsenden an:

Verlagsunion
Friedrich-Bergius-Straße 20
Postfach 57 07
6200 Wiesbaden

Bestellgutschein

Ja, ich möchte ins Computerleben einsteigen und bestelle deshalb:

☐ Jahresabonnement **CPU** (6 Hefte) 60,- DM
☐ Jahresabonnement **HC und Computronic** (12 Hefte) 60,- DM
☐ Jahresabonnement **Kombi: HC, Computronic und CPU** (18 Hefte) 85,- DM
(Ausland: siehe Impressum)

Name/Vorname _____

Straße, Nr. _____ PLZ, Ort _____

Ich wünsche folgende Zahlungsweise:

☐ Bargeldlos durch Bankeinzug _____ Bankleitzahl

Geldinstitut _____ Konto-Nr. _____

☐ gegen Rechnung _____ Datum, Unterschrift

Abonnements-Kündigungen:
6 Wochen vor Ablauf des Jahresabonnements.
Außerdem kann ich diese Bestellung innerhalb von 10 Tagen beim Verlag widerrufen!
Zur Wahrung der Frist genügt die rechtzeitige Absendung des Widerrufs!

_____ Datum, Unterschrift

Bei Ihrem Zeitschriftenhändler erhältlich!

monatlich

Compute mit

COMMODORE & SCHNEIDER

5/85

VC-64, VC-20, C-16, CPC-464

Unabhängiges Magazin für Anwender von Commodore- und Schneider-Computern

2,80 DM
24 öS
2,80 sFr

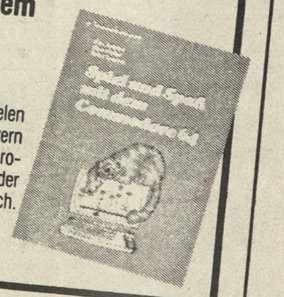
Aktuell +++ Aktuell +++ Aktuell +++ Aktuell

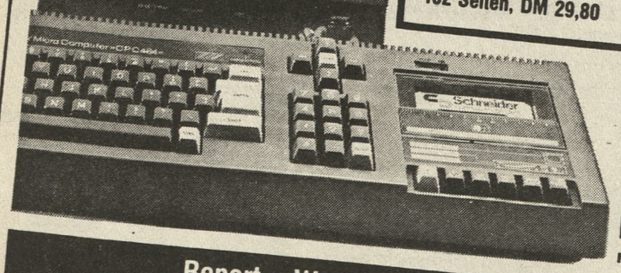
„COMPUTE MIT..." haben wir, lieber Leser, speziell auf die Commodore- und Schneider- (hier der CPC 464)-Computer zugeschnitten. Nachdem sich schon im Herbst 1984 auf dem deutschen Computer-Markt abzuzeichnen schien, daß der CPC 464 nach einem nahezu kometenhaften Aufstieg gute Chancen besitzt, dem Giganten Commodore ein ernstzunehmender Konkurrent zu werden, wurde unser Team nahezu gezwungen, diesen Trend aufzunehmen und auch weiterhin zu verfolgen. D. h., für beide populären Homecomputer-Riesen werden wir in jeder Ausgabe von „COMPUTE MIT..." ein attraktives Angebot bereithalten. Außerdem bietet unsere Redaktion einen Super-Service an, um Ihnen eventuelle schwierige Abtippereien zu ersparen. Mehr darüber auf Seite 11!

Spiel und Spaß mit dem Commodore 64
Ron Jeffries, Glen Fisher & Brian Sawyer

Diese Sammlung von 35 Spielen garantiert Commodore 64-Benutzern viele unterhaltende Stunden. Programmlisten und Erläuterungen der Spiele machen die Eingabe einfach.

ISBN 3-89028-014-5
182 Seiten, DM 29,80





Tips & Tricks für CPC-464

ROM-Calls ohne Einsprungbedingungen
mehr darüber auf Seite 55

Report – Werkstatt – Hardware – Bücher –
Software – Tips & Tricks – Assemblerkurs Teil 1 –
Jungle – Sequenzer – Olympia – Cobra – Krümi – Q-Bert u. v. mehr

STOP +++ Neue Ausgabe +++ STOP

Kleinanzeigen

Liebe Leser,

wollen Sie eine Kleinanzeige aufgeben? Suchen Sie einen gebrauchten Home-Computer, Software, Hardware, Kontakte zu Computer-Clubs etc.?

Dann nutzen Sie jetzt den Kleinanzeigenservice der „CPU": Die „CPU" ist ein kompetentes Anwender-Magazin.

Sie erreichen durch uns eine Vielzahl von Lesern im In- und Ausland.

Und so wird's gemacht:

Kleinanzeigen-Coupon ausfüllen (alle Zeilen können ausgefüllt werden) und **einsenden an**

Tronic-Verlag
Postfach 41
3444 Wehretal 1

Also, schnell ausfüllen und einwerfen in den nächsten Postkasten. Dann ist mit Sicherheit Ihre Kleinanzeige dabei.

- CPU - Basar

Kleinanzeigen

- CPU - Basar

Bitte veröffentlichen Sie ab nächstmöglicher Ausgabe unter der Rubrik:

☐ SOFTWARE ☐ HARDWARE ☐ VERSCHIEDENES

Mein Computersystem: _____

folgenden Text:

Bitte jeweils 28 Buchstaben pro Zeile! Bitte Absender nicht vergessen!

Preis für Kleinanzeigen: 10 DM je Kleinanzeige
(keine gewerbliche (Nutzung)) ☐ per Vorkasse ☐ Scheck ☐ Bar

Name/Vorname:

Straße, Nr.: PLZ/Ort:

Datum, Unterschrift

Ich versichere hiermit, daß die angebotenen Programme/Beiträge frei von Rechten Dritter sind!

Welche Vorteile bringt der Weltstandard für Home-Computer?

- MSX ist der erste praktikable Weltstandard in der Computerbranche, auf den sich alle wichtigen Hardware-Lieferanten von Anfang an geeinigt haben.
  Mit MSX wurde endlich die Grundvoraussetzung für den Durchbruch des Home-Computers in breiten Konsumentenkreisen geschaffen. Denn nur volle System-Kompatibilität macht den Computer im Heim interessant: Programme müssen austauschbar sein, Anschlüsse für Peripheriegeräte wie Drucker, Diskettenlaufwerke, Telefonmodems und Joysticks müssen derselben Norm entsprechen.
- Erfahrene Programmier-Profis und jugendliche Computer-Freaks benutzen alle BASIC. Nur: Heute spricht jeder Home-Computer einen anderen Dialekt.
  „MSX" ist die Abkürzung für „Microsoft Extended Basic", also eine ausgereifte Version von „Microsoft Basic", der weitverbreitetesten Programmiersprache der Welt. Sozusagen das „Hochdeutsch" unter den Computer-Sprachen.
- Natürlich können Sie mit einem MSX-Computer spielen. Aber MSX eignet sich aufgrund seiner aufwärtskompatiblen Struktur auch für viele ernsthafte Anwendungen in Beruf und Büro.
  MSX läßt sich leicht zu CP/M aufrüsten, international eines der bewährtesten Betriebssysteme für Personal-Computer. Das heißt: Dem MSX-Anwender stehen viele Tausend weltweit verbreitete User-Programme zur Verfügung. MSX garantiert den nahtlosen Anschluß an den mittleren PC-Bereich – zum niedrigen Preis!
  Alle sprechen heute von neuen Medien. Btx, Videotext und andere Systeme benötigen ein „Steuerpult", um sinnvoll eingesetzt zu werden. Dafür eignen sich Home-Computer nach dem MSX-Weltstandard besonders gut. Und dank ihrer Vielseitigkeit können sie auch als „Zentrale" für andere Dinge im Haus verwendet werden: HiFi-Anlage, Videogerät oder Hausantenne, zum Beispiel.
- Die meisten MSX-Partner sind erfahrene Elektronik-Hersteller mit reichlich Erfahrung im Bau und Verkauf von Geräten für den Endverbraucher. Hinter ihnen steht

die Marktkraft einer etablierten Branche. MSX wird über die angestammten Vertriebskanäle dieser Firmen zum Verbraucher gelangen.
Gerade der Fachhändler für Rundfunk, Fernsehen, HiFi und Photo hat sich in den letzten Jahren immer wieder auf technologische Neuerungen einzustellen gewußt. Er weiß, wie man Kunden berät und die eigenen Mitarbeiter schult. Zusammen mit dem Computer-Fachhandel wird es also schon bald eine lückenlose Versorgung des Verbrauchers in ganz Deutschland mit Geräten und Software nach dem Weltstandard MSX geben.

- Home-Computer nach dem MSX-Standard sind ideale Lern- und Arbeitsinstrumente gerade für Kinder, Jugendliche und Heranwachsende. Weil MSX erschwinglich ist. Weil MSX unter derzeit mindestens 15 Marken absolut kompatibel ist; ein großer Teil der Schüler wird also den Unterricht zu Hause vertiefen können. Und weil viele Programm-Anbieter demnächst mit „Education"-Software aufwarten werden, die pädagogisch auf den Geschmack und das Lerntempo der jeweiligen Altersgruppen abgestimmt ist.
  Mit MSX werden Schüler beispielsweise in der „Kinder - Programmiersprache" LOGO arbeiten können, wie das von vielen Lehrern gefordert wird. Und MSX wird den nahtlosen Übergang zu MS-DOS gewährleisten – dem Standard in den oberen Computer-Stufen und später in der Wirtschaft.
- MSX gibt jedem Hersteller volle Freiheit, innerhalb des vom Weltstandard vorgegebenen breiten Rahmens Computer und Zusatzgeräte ganz nach dem Wunsch seiner Kunden zu bauen.
  Schon die Tatsache, daß beliebige Peripherie-Geräte mit jedem MSX-Computer laufen, läßt erwarten, daß es auch in Deutschland eine Vielzahl verschiedenster Extras und Zusatzgeräte für jeden Geschmack und Geldbeutel geben wird. Auch die Computer selbst können sehr verschieden aussehen und in sehr unterschiedlichen Preisklassen liegen, je nach Leistung.

Entscheidende Weiterentwicklung in der Halbleiterfertigung

ERSTE GROSSE SILIZIUM-SCHEIBE VON SINCLAIR

Sinclair Research Ltd. hat die ersten kommerziell einsetzbaren großen Siliziumscheiben in seinem Forschungszentrum Metalab in Cambridge produziert. Damit ist dem Unternehmen ein wichtiger Durchbruch im Elektronikbau gelungen.
Zu den Vorteilen des WSI-Verfahrens (Wafer Scale Integration), einer neuen Halbleiter-Fertigungstechnologie, zählen immense Einsparungen an Größe und Kosten sowie erhebliche Verbesserungen in Geschwindigkeit und Energieverbrauch gegenüber gängigen Produkten.
Nach Angaben des Unternehmens ist der Versuch, die ersten Scheiben mit der Sinclair eigenen WSI-Technologie auf einer konventionellen Halbleiter-Produktionslinie in Großbritannien zu fertigen, erfolgreich abgeschlossen. Alle Tests hätten die Erwartungen des Unternehmens erfüllt oder sogar übertroffen und bestätigt, daß sich die Sinclair Silizium-Scheiben für die Massenproduktion eignen.
Die WSI-Technologie, die über ein Jahrzehnt lang von weltweit führenden Computer- und Elektronik-Unternehmen vorangetrieben worden ist, stellt die nächste logische Entwicklung in der Herstellung von zunehmend komplexen elektronischen Komponenten dar. Diese Entwicklung dürfte ebenso bedeutend sein wie die Erfindung von Radioröhren, Transistoren oder der erst kürzlich entwickelten IC Halbleiter-Chips und ist weiterhin von immenser Bedeutung für Hersteller in der Konsumgüter-Elektronik, in der Industrie, Medizin und in der Rüstungsindustrie".
Halbleiterchips werden derzeit auf Siliziumscheiben mit einem Durchmesser von zehn oder zwölf Zentimetern gefertigt. Sie werden aufgeschnitten, um Ausschüsse von den hunderten funktionsfähiger Chips zu trennen. Die Chips müssen dann individuell montiert, verdrahtet und in Plastik verpackt werden, bevor sie auf vergleichsweise großen Schaltungsplatten (Printed Circuit Boards) zusammengesetzt werden.
Mit der WSI-Technologie läßt sich die Scheibe anstelle einer Schaltungsplatte benutzen. Fehlerhafte Chips werden von in dieser Scheibe eingebauter, elektronischer Logik übergangen, anstatt sie auseinanderzuschneiden. Viele verschiedene Methoden sind ausprobiert worden, um die WSI-Technologie einzusetzen. Bis heute hat keine zur Kommerzialisierung eines Produktes geführt – entweder waren die Methoden zu teuer oder brachten einen zu geringen Ertrag an funktionsfähigen Chips pro Scheibe.
Mit der Sinclair Methode werden Schaltkreise auf der Scheibe selbst verwendet, um fehlerhafte Chips zu identifizieren und automatisch zu isolieren. Vom britischen Wissenschaftler Ivor Catt in den frühen siebziger Jahren erfunden, ist es einem Team von Metalab gelungen, diese Technik in einem Produktionsumfeld zum Arbeiten zu bringen. Die bisher getätigten Investitionen betrugen mehr als 1 Mio. Pfund.

Das erste Sinclair WSI-Produkt, ein 1/2 MBit Speicherchip für den Sinclair QL Computer, wird im konventionellen NMOS - Halbleiter - Verfahren hergestellt. Das Unternehmen ist gemäß eigenen Angaben in der bipolaren Technologie, die weitere größere Reduktionen in Größe und Kosten bringen wird, sehr weit fortgeschritten.

Computer – Kurse der Computerschule Waldbrunnen

Sehr geehrte Leser! An dieser Stelle weisen wir Sie auf einige Kurse der Computerschule Waldbrunnen hin. Interessierte wenden sich an:

Computerschule Waldbrunnen GmbH,
Brunnenstraße 7,
5461 Windhagen-Rederscheid,
Tel.: 02271-65612

MULTIPLAN
Eintägiges Chef-Seminar

Sie müssen keine Programmiersprache beherrschen, um mit Multiplan auch umfangreichste Tabellenkalkulationen durchführen zu können.
Sie werden nach diesem Seminar wissen, ob sich der Einsatz in Ihrem Unternehmen lohnt.
Auch wenn Sie bereits Multiplan besitzen, erhalten Sie sicherlich wertvolle Tips für den Einsatz.
Teilnehmer:
Mittelständler, Freiberufler, Manager
max. 12 Teilnehmer
jew. zwei Teilnehmer arbeiten an einem PC

Voraussetzungen:
Grundkenntnisse des Arbeitens mit dem PC z. B. Grundkurs im Rahmen des Waldbrunnen-Computer-Führerscheins

Dauer: 1 Tag
Kosten: DM 250,–
incl. Teilnehmerunterlagen

Termine zur Auswahl:
26. 08. 1985
21. 09. 1985
15. 10. 1985
19. 11. 1985
14. 12. 1985

Referenten:
Ein Informatik-Lehrer von einem NRW-Gymnasium, der Geschäftsführer der Computer-Schule Waldbrunnen, ein Vertreter des Bundesministeriums für Bildung und Wissenschaft (beim Kamingespräch).
Es werden keine EDV-Kenntnisse vorausgesetzt.

Lehrer – Computer – Camp vom 7.-11. 10. 85

Sie sind Lehrer an einer allgemeinbildenden Schule? Sie werden von allen Seiten bedrängt, doch endlich Computer im Unterricht einzusetzen (nicht nur an Gymnasien im Fach Informatik)?
Sie möchten ja schon wollen, aber können dürfen Sie nicht? Ihnen fehlen nicht nur Erfahrungen im Umgang mit Computern im Unterricht, vor allen Dingen wissen Sie nicht, woher Sie die Geräte nehmen sollen?
In wenigen Jahren werden die ersten I-Dötze mit Schultüte und vielleicht bemerkenswerten LOGO-Kenntnissen antreten. Was werden Sie mit denen anfangen?
Wir möchten mit Ihnen ins Gespräch kommen.
Im Lehrer-Computer-Camp vom 7.-11. 10. 85 werden wir nicht nur Grundkenntnisse der Programmiersprache BASIC erarbeiten. Wir werden Ihnen die Besonderheiten von LOGO und PASCAL verdeutlichen und Einsatzmöglichkeiten der kleinen Rechner an Schulen aufzeigen.
Das Camp findet in den NRW-Herbstferien statt. Wir werden 6 U-Std. täglich an Computern arbeiten (jeweils zu dritt am Schneider CPC 464).
Die Teilnehmerzahl ist auf 30 begrenzt.
Die Teilnehmergebühr beträgt DM 200,– pro Person incl. MWSt., zuzügl. Hotelkosten.

Eintägiges Chef-Seminar

Am Beispiel des Textprogrammes Wordstar werden Ihnen die Möglichkeiten zeitgemäßer Textverarbeitung am Personal-Computer demonstriert.
Sie werden nach diesem Seminar erkennen, ob sich der Einsatz in Ihrem Unternehmen lohnt.
Auch wenn Sie bereits Wordstar besitzen, erhalten Sie sicherlich wertvolle Tips für den Einsatz.
Dauer: 1 Tag
Kosten: DM 250,– incl. Teilnehmerunterlagen

Impressum
Tips + Tricks v. CPU erscheint alle 2 Monate im Tronic-Verlag, 3444 Wehretal 1

Redaktion:
Axel Credé (verantwortlich)
Chefredakteur Siegfried Görk,
Frank Brall, Harmut Wendt, Holger Crede,
Manfred Kleimann, Ottfried Schmidt

Freie Mitarbeiter
Volker Becker, Rolf Freitag

Gesamtherstellung:
Druckhaus Dierichs Kassel, Frankfurter Straße 168, 3500 Kassel

Vertrieb:
Inland (Groß-, Einzel- und Bahnhofsbuchhandel) sowie Österreich und Schweiz
Verlagsunion
Friedrich-Bergius-Straße 20, 6200 Wiesbaden
Telefon 06121/2660

Anfragen nicht an Vertrieb oder Druckerei, sondern nur an den Verlag!

Anzeigenleitung:
Markt-Kommunikation Anzeigenverwaltung GmbH + Co KG, Hohenzollernstr. 60, 8000 München 40, Telefon 089/334013, Telex 0529212
BTX-Leitseite *31880#

Erscheinungsweise:
Erstverkaufstag von CPU ist Mitte des Monats.

Anzeigenpreise:
Bitte Mediaunterlagen anfordern.

Anzeigenannahmeschluß:
Jeweils 3 Wochen vor Erscheinungstermin.

Urheberrecht:
Alle in CPU veröffentlichten Beiträge sind urheberrechtlich geschützt. Alle Rechte, auch Übersetzungen, vorbehalten.
Reproduktionen jeder Art (Fotokopien, Microfilm, Erfassung in Datenverarbeitungsanlagen usw.) bedürfen der schriftlichen Genehmigung des Verlags. Alle veröffentliche Software wurde von Mitarbeitern des Verlages oder von freien Mitarbeitern erstellt.
Aus ihrer Veröffentlichung kann nicht geschlossen werden, daß die beschriebenen Lösungen oder Bezeichnungen frei von Schutzrechten sind.

Bezugspreis:
Einzelheft 6,– DM
Abonnements:
CPU

Inland	DM 60,––
Europäisches Ausland	DM 87,50
Außereuropäisches Ausland	DM 120,––

Kombi:
CPU + Homecomputer + Computronic

Inland	DM 85,––
Europäisches Ausland	DM 127,50
Außereuropäisches Ausland	DM 170,––

Autoren, Manuskripte:
Der Verlag nimmt Manuskripte und Software zur Veröffentlichung gerne entgegen.
Sollte keine andere Vereinbarung getroffen sein, so gehen wir davon aus, daß Sie mit einem Honorar von 120,– DM pro abgedruckter Seite im Heft einverstanden sind.
Bei Zusendung von Manuskripten und Software erteilt der Autor dem Verlag die Genehmigung zum Abdruck und Versand der veröffentlichten Programme auf Datenträger.
Rücksendung erfolgt nur gegen Erstattung der Unkosten. Zusendungen von Software zur Veröffentlichung sollten folgendes enthalten:
Kopierfähige Kassette oder Diskette mit dem Programm (Computer-Bezeichnung), von Drucker erstelltes Listing oder Serie von Bildschirmfotos (keine Schreibmaschinenlistings), evtl. Bildschirmfotos von einem Probelauf und ausführliche Programmbeschreibung (Erklärung der Variablenliste, Beschreibung des Bildschirmaufbaues, Farbe, Grafik usw). Für eingesandte Programmunterlagen kann keinerlei Haftung übernommen werden.

TRONIC-SOFTWARE-SERVICE

Atari, Mastermind/Schlangenkrieg, Kassette 10,50 DM, Diskette 19,50 DM, Bestell-Nr. I 41

Tank-Battle/Oil Panic, Kassette 12,50 DM, Bestell-Nr. I 51

Startup / Zeilen - Split / Tomstone-City, Diskette 19,50 DM, Bestell-Nr. I 71

Painter/Hardcopy, Kassette 14,– DM, Diskette 19,50 DM, Bestell-Nr. I 81

The Big Quest/Fünf gewinnt, Kassette 14,– DM, Diskette 19,50 DM, Bestell-Nr. I 91

Splitt, Kassette 11,– DM, Diskette 17,50 DM, Bestell-Nr. I 101

Ski/Mutation, Kassette 14,50 DM, Diskette 19,50 DM, Bestell-Nr. I 121

Super Miner / Diamonds, Kassette 14,50 DM, Diskette 19,50 DM, Bestell-Nr. I 22

Donkey Kong/Kerzenheinz, Kassette 16,50 DM, Diskette 19,50 DM, Bestell-Nr. I 32.

TI-99, Drei - Kronen - Spiel / Zahlenputzen, Kassette 8,50 DM, Bestell-Nr. T 41

Karl der Käfer/Alien-Landing, Kassette 14,50 DM, Bestell-Nr. T 51

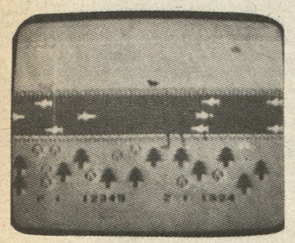
Jack the Digger I/Noah 2099, Kassette 14,50 DM, Bestell-Nr. T 61

Lift Bär/ASC II DEF Teil 1, Kassette 14,– DM, Bestell-Nr. T 71

Maya/ASC II DEF Teil 2, Kassette 14,50 DM, Bestell-Nr. T 81

ASC II DEF Teil 1 + 2, Diskette 19,50 DM, Bestell-Nr. T 881

Transfer/Silverspar, Kassette 14,50 DM, Diskette 19,50 DM, Bestell-Nr. T 91



Mother Duck//Screen Designer, Kassette 14,50 DM, Diskette 19,50 DM, Bestell-Nr. T 101

Cave Man, Kassette 8,– DM, Diskette 11,50 DM, Bestell-Nr. T 121

Moon Race/Frogger/Slicks, Kassette 19,50 DM, Diskette 25,50 DM, Bestell-Nr. T 22

Panzerschlacht, Kassette 8,– DM, Diskette 11,50 DM, Bestell-Nr. T 32.

Commodore 64, Mauern/Widerstand, Kassette 8,– DM, Diskette 15,– DM, Bestell-Nr. C 41

Space-Comets / Erdspalte / Sprite-Data, Kassette 15,– DM, Diskette 23,50 DM, Bestell-Nr. C 51

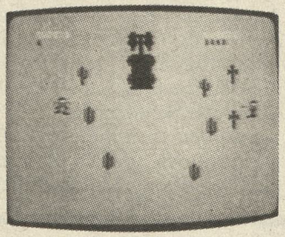
Autostart/Bestellschein/Roadpainter, Kassette 16,50 DM, Diskette 23,50 DM, Bestell-Nr. C 61

Hardcopy / Space-Fighter / Data-Generator, Kassette 15,50 DM, Diskette 19,50 DM, Bestell-Nr. C 71

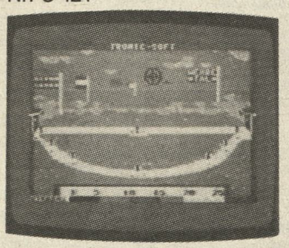
Monster-Attack/Block-Painter/Epson-Drucker, Kassette 16,50 DM, Diskette 23,50 DM, Bestell-Nr. C 81

Projekt / Datenbank, Kassette 16,– DM, Diskette 23,50 DM, Bestell-Nr. C 91

Spiders/The Basic, Kassette 16,50 DM, Diskette 23,50 DM, Bestell-Nr. C 101



High Noon / Skeet / Grafik-Designer Kassette 17,50 DM, Diskette 23,50 DM, Bestell-Nr. C 121



Painter/Star-Baddle/Editor, Kassette 17,50 DM, Diskette 23,50 DM, Bestell-Nr. C 22

Wüstenrallye/Jet-Pac/Black Moore Castle Kassette 17,50 DM, Diskette 23,50 DM, Bestell-Nr. C 32.

VC-20, Bestellschein /Glücky, Kassette 8,50 DM, Diskette 16,– DM, Bestell-Nr. V 61

Multigraph / All Rammer, Kassette 11,– DM, Diskette 15,50 DM, Bestell-Nr. V 71

Zyklo/Meteorit, Kassette 11,– DM, Diskette 15,50 DM, Bestell-Nr. V 81

Garten / Schloß Gruselstein, Kassette 14,– DM, Diskette 19,50 DM, Bestell-Nr. V 91

Fressman/Outlaw, Kassette 14,– DM, Diskette 19,50 DM, Bestell-Nr. V 101

Prost/Buffalo Bill, Kassette 14,– DM, Diskette 19,50 DM, Bestell-Nr. V 121

Joy Man/Powerpack, Kassette 14,– DM, Diskette 19,50 DM, Bestell-Nr. V 22

Der rasende Malocher/Frankie goes to Pharao, Kassette 14,– DM, Diskette 19,50 DM, Bestell-Nr. V 32.

Apple II, Wilder Westen / Karambolage / Maskengenerator, Diskette 19,50 DM, Bestell-Nr. A 41

Music-Maker/Mission: Adler/Disk-Katalog, Diskette 19,50 DM, Bestell-Nr. A 51

Snake/Super Datei/Shapetables, Diskette 19,50 DM, Bestell-Nr. A 61

Library/Fight, Diskette 19,50 DM, Bestell-Nr. A 71

Reversal / Disk - Menue - Generator, Diskette 19,50 DM, Bestell-Nr. A 81

Diamonds / Hilfsprogramm, Diskette 19,50 DM, Bestell-Nr. A 91

Tic - Tac - Toe / Jumper, Diskette 19,50 DM, Bestell-Nr. A 101

Donovan / Basic - Konverter, Diskette 19,50 DM, Bestell-Nr. A 121

Funktionstasten / Painter / Bowling, Diskette 19,50 DM, Bestell-Nr. A 22

Thundet/Castle of Doom, Diskette 19,50 DM, Bestell-Nr. A 32.

Dragon 32, Blizzard, Kassette 8,– DM, Bestell-Nr. D 41

Space-Flight / Geosoft, Kassette 10,– DM, Bestell-Nr. D 51

Waregames, Kassette 8,50 DM, Bestell-Nr. D 61

Laser-Attack, Kassette 8,50 DM, Bestell-Nr. D 71

Hardcopy, Kassette 13,– DM, Bestell-Nr. D 81

Anwenderprogramm, Kassette 10,– DM, Bestell-Nr. D 91

Dragon Paint, Kassette 14,– DM, Bestell-Nr. D 101

ZX-Spectrum, Inventur, Kassette 12,– DM, Bestell-Nr. S 41.

Missile-Comment, Kassette 8,50 DM, Bestell-Nr. S 51

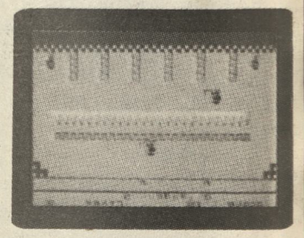
Defender/Lui der Wurm/Alternativer Zeichensatz, Kassette 13,50 DM, Bestell-Nr. S 61

Matheprogramm / Bongo-Beecatcher, Kassette 12,50 DM, Bestell-Nr. S 71

Solitaire / Superstat / Kleinstes gem. Vielfache, Kassette 14,50, Bestell-Nr. S. 81

Jump about, Kassette 14,50 DM, Bestell-Nr. S 91

Pac-Man/Oil Panic, Kassette 14,50 DM, Bestell-Nr. S 101



Frogger, Kassette 16,– DM, Bestell-Nr. S 121

Jump, Kassette 14,50 DM, Bestell-Nr. S 22

Jet Set Freddie, Kassette 8,– DM, Bestell-Nr. S 32.

ZX-81, Go-Ball/Grand/Prix, Kassette 10,– Bestell-Nr. Z 51

Moon-Crash/ZX-Draw, Diskette 10,– DM, Bestell-Nr. Z 61

Tonprogramm / Aldebaran, Kassette 10,– DM, Bestell-Nr. Z 71

Reversi, Kassette 10,– DM, Bestell-Nr. Z 91

Panik Labyrinth, Kassette 10,– DM, Bestell-Nr. Z 101

Expedition, Kassette 10,– DM, Bestell-Nr. Z 121

Spinnen, Kassette 14,50 DM, Bestell-Nr. Z 22

Spukhaus, Kassette 14,50 DM, Bestell-Nr. Z 32.

CPU-SOFTWARE-SERVICE

Commodore 64

***Multi*-Key / S-Tool 64 / Interrupt-Programme,** Kassette 16,– DM, Diskette 21,– DM, Bestell-Nr. UC 2/85

Spritehilfe / Diskloader / Directory, Diskette 21,– DM, Bestell-Nr. UC 3/85.

Tape-Directory / Asmon / Data-Generator, Kassette 15,– DM, Diskette 21,– DM, Bestell-Nr. UC 3/85.

Fast-Load, Kassette 10,– DM, Bestell-Nr. UC 4/85.

Diskmonitor / Zeichensatz, Diskette 20,– DM, Bestell-Nr. UC 5/85.

Reassembler / Maskengenerator, Kassette 15,– DM, Diskette 20,– DM, Bestell-Nr. UC 6/85.

Del 64 / Treiberprogramm für 1526 / *MPS* 802, Decelerator, Kassette 14,– DM, Diskette 19,–DM, Bestell-Nr. UC 7/85

Video-Utilities, Multi-Basic V3, Bildeditor, Kassette 17, DM, Diskette 23,– DM, Bestell-Nr. UC 9/85.

Schneider CPC-464

Autorennen, Kassette 11,– DM, Bestell-Nr. US 3/85.

Universal-Datei V1, Kassette 9,– DM, Bestell-Nr. US 4/85.

Computerschrift / Symbol-Swap, Kassette 10,– DM, Bestell-Nr. US 5/85.

Keyboard Toolkit / Farbdekodierung, Kassette 11,– DM, Bestell-Nr. US 6/1/85.

Aladin, Kassette 9,– DM, Bestell-Nr. US 6/2/85.

Exklusive für alle CPU-Leser

Painter / Box-Befehl, Kassette 14,– DM, Diskette 24,– DM, Bestell-Nr. US 7/85.

Basic, Maschinen-Kit, Kassette 12,– DM, Diskette 22,– DM, Bestell-Nr. US 9/85.

VC-20

Sprites/Space-Battle, Kassette 13,– DM, Diskette 19,– DM, Bestell-Nr. UV 2/85.

Grafik-Painter, Kassette 10,– DM, Bestell-Nr. UV 5/85.

Vic-Clock, Kassette 9,– DM, Bestell-Nr. UV 6/85.

Decelerator, Kassette 9,– DM, Bestell-Nr. UV 7/85.

Disassembler, Old-Programm, Kassette 10,– DM, Bestell-Nr. UV 9/85.

Apple

Asteroids, Diskette 10,– DM, Bestell-Nr. UA 2/85.

Dateiverwaltung, Diskette 18,– DM, Bestell-Nr. UA 3/85.

Digiter / Reset-Saver, Diskette 18,– DM, Bestell-Nr. UA 4/85.

Assembler-Monitor, Diskette 17,– DM, Bestell-Nr. UA 5/85.

Disk Check IC/Dos Change, Diskette 18,– DM, Bestell-Nr. UA 6/85.

Hello, Diskette 17,– DM, Bestell-Nr. UA 7/85.

Digital-Sound-Editor, Basic-Befehle, Diskette 18,– DM, Bestell-Nr. UA 9.85.

ZX-81

The Castle of Questionmark, Kassette 10,– DM, Bestell-Nr. UX 2/1/85.

MSX

Zeichensatzkonverter, Kassette 10,– DM, Bestell-Nr. UM 3/85.

Zirkus, Kassette 13,– DM, Bestell-Nr. UM 5/85.

Pipeline, Kassette 13,– DM, Bestell-Nr. UM 4/85.

Alien Attack, Kassette 13,– DM, Bestell-Nr. UM 7/85.

Typer 60, Disketten-Monitor, Diskette 5¼ Zoll 25,– DM, Kassette (nur für 3 ½ Zoll Diskettenlaufwerk verwendbar) 19,–DM.

TI-99

Olympics, Kassette 11,– DM, Bestell-Nr. UT 2/2/85.

Funktionsanalyse / Hardcopy, Kassette 11,– DM, Bestell-Nr. UT 3/85.

Invaded by the Empire, Kassette 11,– DM, Bestell-Nr. UT 4/85.

Multicolor / Load - Programm, Kassette 14,– DM, Diskette 19,– DM, Bestell-Nr. UT 6/85.

Disassembler, Diskette 19,– DM, Bestell-Nr. UT 7/85.

Superfile, Kassette 11,– DM, Bestell-Nr. UT 9/85.

ZX-Spectrum

Panik Sam / Color Change / Clock, Kassette 13,– DM, Bestell-Nr. UZ 2/85.

MC - Routinen / Data - Base / Pixel, Kassette 11,– DM, Bestell-Nr. UZ 3/85.

Reversi/UDG-Editor, Kassette 15,– DM, Bestell-Nr. UZ 4/85.

Star-Basic, Kassette 13,– DM, Bestell-Nr. UZ5/85.

Fill-Programm/Copy it, Kassette 14,– DM, Bestell-Nr. UZ 6/85.

Trace-Single Step, Variable List, X-BASIC, Kassette 16,– DM, Bestell-Nr. UZ 9/85.

Atari

Mauerflucht / Lunar - Simulation, Kassette 13,– DM, Diskette 19,– DM, Bestell-Nr. UI 2/85.

Ram - Tester / Cherry - Harry, Kassette 13,– DM, Diskette 19,–DM, Bestell-Nr. UI 3/85.

Crazy Egon, Kassette 13,– DM, Bestell-Nr. UI/85.

Monitorprogramm, Kassette 11,– DM, Bestell-Nr. UI 6/85.

Zeicheneditor, Diskette 19,– DM, Bestell-Nr. UI 7/85.

Grafikausgabe, Error-Helper, Diskette 18,– DM, Bestell-Nr. UI 9/85.

ELEPHANT.™ KOMPROMISSLOS ERSTE WAHL.

ELEPHANT MEMORY SYSTEMS™

"NEVER FORGETS."

Speichern und Drucken. Fehlerlos. Mit 100 % Sicherheit: das ist Elephant Memory Systems™.
Unter diesem Markennamen erhalten Sie hochwertiges Zubehör für perfekt arbeitende Kommunikations-Systeme der Text- und Datenverarbeitung – wie Disketten, Farbbänder, Farbbandkassetten und Spezialreinigungsmittel.
Kompromiß ist Risiko. Zweite Wahl ist Unsicherheit – deshalb: verlangen Sie Elephant Memory Systems™.

Dennison

ELEPHANT™ NEVER FORGETS

MARCOM Computerzubehör GmbH
Podbielskistraße 321, 3000 Hannover 51, Tel. (05 11) 64 74 20

Frankreich: Soroclass, 8, Rue Montgolfier · 93115, Rosny-Sous-Bois, Tel.: 16 (1) 855-73-70
Großbritannien: Dennison Mfg. Co. Ltd., Colonial Way, Watford WD2 4JY, Tel.: 0923 41244, Telex: 923321
Italien: King mec s.p.a. · Via Regio Parco 108 · 10036 Settimo Torinese
Weiteres Ausland: Dennison International Company, 4006 Erkrath 1, Matthias-Claudius-Straße 9, Telex: 858 6600